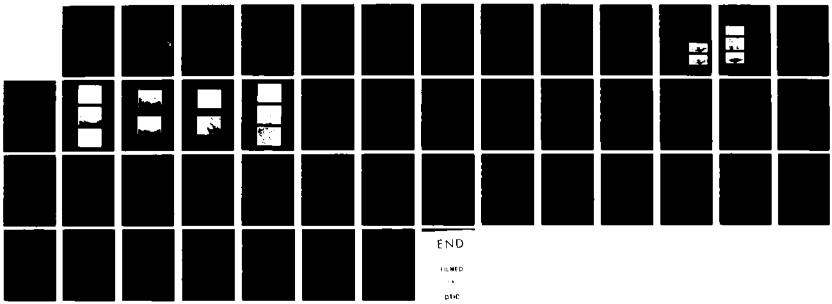
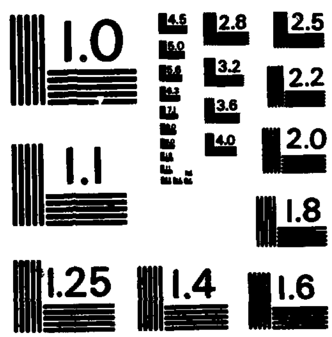
AD-A121 999 STATISTICAL IMAGE PROCESSING(U) SOUTHEASTERN MASSACHUSETTS UNIV NORTH DARTMOUTH DEPT F ELECTRICAL ENGINEERING C H CHEN 16 NOV 82 N00014-79-C-0494 1/1

UNCLASSIFIED F/G 12/1 NL






MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

Final Report

Contract No. N00014-79-C-0494
Task No. NR042-422

November 16, 1982



# STATISTICAL IMAGE PROCESSING

Submitted to:

Statistics and Probability Program
Office of Naval Research, Code 436
Arlington, VA 22217



Prepared by:

Professor C. H. Chen (Principal Investigator)
Dept. of Electrical and Computer Engineering
Southeastern Massachusetts University
North Dartmouth, MA 02747



Unclassified

Key words: image models, image segmentation, maximum entropy spectral analysis, texture image analysis and classification, image software package, automatic spatial clustering.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. AD-A121 999 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) Statistical Image Processing -- Final Report | | 5. TYPE OF REPORT & PERIOD COVERED Final Report |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) C. H. CHEN | | 8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0494 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Electrical and Computer Engineering Dept. Southeastern Massachusetts University N. Dartmouth, Mass. 02747 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR042-422 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Statistics and Probability Program Office of Naval Research Arlington, VA 22217 | | 12. REPORT DATE Nov. 16, 1982 |
| | | 13. NUMBER OF PAGES |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

see cover page

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see cover page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

STATISTICAL IMAGE PROCESSING

Final Report
(for period June 1, 1979 to August 31, 1982)

Abstract

This report summarizes the major research work accomplished under the contract including the topics of statistical image models, comparative evaluation of image processing techniques, image segmentation algorithms, two-dimensional maximum entropy spectral analysis, and spatial clustering algorithms with applications to artificial and remotely sensed images. Detailed list of publications available in open literature is provided. A list of software package generated is included in the Appendices.

## I. Introduction

This report is organized according to the topics we have worked under this Contract. A brief summary of each is presented. Detailed list of publications is provided in References. Over 40 technical reports prepared under the Contract are not listed however as most results documented in reports were published as listed in References. Copies of two papers are included in the Appendices. A detailed list of image processing software package generated is also included in the Appendices. The program listings in magnetic tapes were delivered to Dr. Doug DePriest in October 1981.

## II. Major Research Results

1. Statistical image processing techniques for additive noise case were compared. Median filtering followed by Kalman adaptive filtering is most effective. For Seasat images the multicative noise removal is considered by using local statistics. (Refs. 16,20)

2. Statistical image segmentation studies include the extensive comparative evaluation of supervised and unsupervised segmentation techniques for texture and infrared images. The segmentation is performed by pixel classification. Both Fisher's linear discriminant and the maximum a posteriori estimation procedures are found to be very effective. Statistical techniques however are limited to pixel based segmentations. (Refs. 4,9,10,12,13,14)

3. Statistical image modeling study is concerned with the auto-regressive models and low order ARMA models. Such modeling leads to image enhancement, segmentation and classification. These models provide a nice way to take into account the contextual dependence

among the nearest neighbors. The question remains whether the object boundary should have a separate model from the remaining homogeneous parts of the image. (Refs. 15,18)

4. An automatic spatial clustering algorithm has been developed for image segmentation and compression. The algorithm can determine the minimum number of clusters and can also work with a specified number of clusters. The algorithm has been successfully tested with various images including USC image data base, Seasat images, and U.S. Army topographic images. (Refs. 3,4)

5. A two-dimensional maximum entropy spectral analysis algorithm was thoroughly developed and tested for texture image analysis, classification, segemntation as well as general purpose spectral computation based on limited number of data points. (Refs. 1,2,4,5, 6,8)

6. An initial effort of tracking image sequence was made by using pixel classification for object extraction. Further study to model the statistics of image variation is much needed. (Ref. 4)

## III References

The following is a list of publications by C.H. Chen with full or partial support of Contract N00014-79-C-0494.

1. "A study of texture classification using spectral features", Proc. of the 6th INternational Conference on Pattern Recognition, pp.1074-1077, Munich, Germany, Oct. 19-22, 1982.
One copy of the paper is attached as Appendix B of the report.

2. "Nonlinear Maximum Entropy Spectral Analysis Methods for Signal Recognition", book, Research Studies Press, England, already published in October, 1982.

3. "On a spatial clustering algorithm for image analysis", in Proc. of NATO Advanced Research Workshop on Issues in Acoustic Signal/ Image Processing and Recognition, held in San Miniato, Italy, Aug. 5-9, 1982, and to be published by Springer-Verlag, Heidelberg, Germany, January 1983, pp.309-314.
One copy of the paper is attached as Appendix C of the report.

4. "Efficient algorithms for digital processing of remotely sensed imagery", presented at ONR Workshop on Signal Processing in the Ocean Environment, Annapolis, Md., May 11-14, 1982. Full paper is to be published in a proceedings volume.

5. "Digital Waveform Processing and Recognition", book, mostly authored by C.H. Chen, CRC Press, Boca Raton, FL, January 1982.

6. "On a two-dimensional maximum entropy spectral estimation method for the texture-image analysis", presented at the Computer Science and Technology Conference, June 4-6, 1982 in Newton, MA.

7. "Adaptive and learning algorithms for seismic detection of personnel", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-4, no. 2, pp. 129-132, March 1982.

8. "Spectral resolution of Fougere's maximum entropy spectral analysis", Proc. of the IEEE, vol 69, no. 7, pp.839-841, July 1981.

9. "On the statistical image segmentation techniques", Proc. of the IEEE Conference on Image Processing and Pattern Recognition, pp. 262-266, August 1981.

10. "A comparison of image segmentation techniques", Proc. of International Conference on Cybernetics and Society, Atlanta, GA, pp. 364-368, October 1981.

11. "A review of geophysical signal analysis and recognition", Proc. of the 2nd International Symposium on Computer-aided Seismic Analysis and Discrimination, pp. 144-152, August 1981.

12. "Statistical image segmentation", American Mathematical Society Regional Meeting Special Session on Cluster Analysis, Oct. 17, 1981 at University of Massachusetts, Amherst, MA.

13. "Comparative evaluation of statistical image segmentation techniques", presented at the American Statistical Association annual conference in Detroit, Michigan, August 1981.

14. "Object isolation with FLIR images using Fisher's linear discriminant", Pattern Recognition journal, vol. 15, pp. 153-159, May 1982.

15. "On two-dimensional ARMA models for image analysis", Proc. of the 5th International Conference on Pattern Recognition, Miami Beach, FL, pp. 1128-1131, Dec. 1980.

16. "A comparison of statistical image processing techniques", Proc. of the IEEE International Conference on Cybernetics and Society, pp. 557-560, Oct. 1980.

17. "Learning in statistical pattern recognition", Proc. of the IEEE International Conference on Cybernetics and Society, pp. 924-929, Oct. 1980.

18. "Statistical image models", presented at the Eastern Regional Conference of American Statistical Association and Biometrics Society, Charlestown, S. Carolina, March 1980.

19. "Applications of statistical pattern recognition", Proc. of the Statistical Computing Section of the American Statistical Association, Dec. 1979.

20. "Statistical image processing and recognition", Proc. of the IEEE COMPSAC, pp. 64-68, November 1979.

## Appendix A Master's Theses

The following is a list of master's theses completed under full or partial support of the Contract, under Prof. Chen's supervision.

1. W.H. Yang, "Automatic spatial clustering and target tracking", July 1982.
2. G.K. Young, "Maximum entropy spectral analysis methods with application to image recognition", May 1982.
3. S.J. Chern, "Time delay estimation and target motion analysis", May 1982.
4. P.G. Ho, "Statistical image segmentation by computer", July 1981.
5. R.H. Wu, "Statistical image segmentation and seismic analysis", May 1981.
6. C. Yen, "Two new approaches to statistical signal and image processing", April 1981.
7. C.Y. Hu, "Study of a class of computer vision problems", August, 1980.
8. J. Chen, "Statistical techniques in seismic signal detection and reconnaissance image analysis", May 1980.

# Appendix B

## A STUDY OF TEXTURE CLASSIFICATION USING SPECTRAL FEATURES

C. H. Chen

Electrical Engineering Department
Southeastern Massachusetts University
N. Dartmouth, Mass. 02747 U.S.A.

### Abstract

Effort in the past on the use of spectral features for discrimination of texture images has had limited success and the feature measures computed from the co-occurrence matrix are preferred. In this paper the superiority of spectral features for texture classification is demonstrated. A new two-dimensional maximum entropy spectral analysis is developed which provides superior resolution capability. Thus accurate power spectrum can be determined from which various ring and wedge spectral features are computed in polar coordinates. Extensive computer results reported indicate that the spectral features so computed provide not only a good measure of texture coarseness and directionality, but also comparable or better classification performance than that reported earlier. A typical performance of over 80 percent correct classification is available from the extracted spectral features by using the Fisher's linear discriminant for classification. A set of normalized features which use both ring and wedge features is particularly recommended. Computationally the method described in this paper is far better than the use of co-occurrence matrix as the iterative algorithm used for spectrum estimation is very fast, even with the use of a minicomputer.

### I. Introduction

Although it is generally recognized that texture images contain statistical, spectral and structural domain information, the use of spectral information alone can be quite effective in the texture-image analysis studies such as texture discrimination and segmetnation. Bajcsy and Liberman [1] expressed the power spectrum in polar coordinates, then integrate over r and $\phi$ to obtain the two-dimensional functions. The location of peaks in these functions indicates prominant texture coarseness and directionality. Weszka et.al. [2] integrated the power spectrum within 16 spatial frequency scales which were combinations of four 1-octave frequency ranges and four $45^{o}$ orientation sectors. They also computed eight "contrast" measures based on the cooccurrence matrix, and obtained better discrimination than with the power spectrum measures. Laws [3] computed a number of energy measures by filtering the texture with sets of small linear operators, then squaring and summing the output of each filter. He reported better discrimination with the energy than with the co-occurrence measures.

A fundamental problem with the power spectrum analysis is the computational accuracy and computational complexity. For texture study, accurate power spectrum must be computed from the small image segments. In this case, the two-dimensional Fourier analysis cannot provide sufficient accuracy as the Fourier analysis is more accurate with a large number of pixels. The two-dimensional maximum entropy spectral analysis, however, is very suitable for a small number of pixels. The computational complexity has been a drawback in using the two-dimensional maximum entropy spectral estimation methods. Recently, Lim and Malik [4] have proposed an efficient iterative algorithm for the two-dimensional maximum entropy power spectrum estimation which can obtain good resolution and sufficient accuracy for the finite sample two-dimensional data. A study of the spectral estimation of texture image has been proved to be successful [5] by using a minicomputer. In this paper, this method is used for the calculation of spectral features of texture image. In section II, the two-dimensional maximum entropy power spectrum estimation is briefly discussed. The method of selection of features will be described in section III while section IV provides some experimental results of the textural classification.

### II. Two-Dimensional Maximum Entropy Power Spectrum Estimation

The basic concept of the maximum entropy method (MEM) of spectral estimation is to extrapolate the autocorrelation function of a random process by maximizing the entropy H of the corresponding probability density function

$$H = \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} \log \hat{P}_x(\omega_1, \omega_2) d\omega_1 d\omega_2 \qquad (1)$$

where $\hat{P}_x(\omega_1, \omega_2)$ is the power spectrum estimate of the random process $x(n_1, n_2)$. The characteristics of this method are equivalent to the autoregressive signal modeling and the power spectrum is calculated by



CH1801-0/82/0000/1074 $00.75 © 1982 IEEE

$$\hat{P}_x(\omega_1, \omega_2) = \frac{1}{\sum\limits_{(n_1,n_2)\epsilon A}\sum \lambda(n_1,n_2)e^{-j\omega_1 n_1} e^{-j\omega_2 n_2}} \quad (2)$$

where $\lambda(n_1, n_2)$ is the autocorrelation whose power spectrum is $1/\hat{P}(\omega_1, \omega_2)$ and A is a set of points $(n_1, n_2)$ where the autocorrelation is known.

Since the filter coefficients cannot be obtained directly by solving the normal equation as in the one-dimensional case, Lim and Malik developed a new iterative algorithm, using adaptive filtering concepts. The basic idea of this algorithm is on the notion that the given correlation point in region A is consistent and the corresponding coefficient should be zero outside region A and proceed this iteration repeatedly until an optimal solution is obtained. That is, for a given $R_x(n_1, n_2)$ for $(n_1, n_2)\epsilon A$, determine $P_x(\omega_1, \omega_2)$ such that $\hat{P}_x(\omega_1, \omega_2)$ satisfy (2) and

$$R_x(n_1, n_2) = F^{-1}\left[\hat{P}_x(\omega_1, \omega_2)\right] \text{ for } (n_1, n_2)\epsilon A$$

A simple flowchart is shown in Fig. 1. We begin with some initial estimate of $\lambda(n_1, n_2)$, obtain the corresponding correlation function, correct the resulting correlation for $(n_1, n_2)\epsilon A$ with the known $R_x(n_1, n_2)$, obtain the corresponding $\lambda(n_1, n_2)$ from the correct corrleation function, and then replace the resulting $\lambda(n_1, n_2)$ with zero for $(n_1, n_2)\epsilon A$. This completes one iteration and the corrected $\lambda(n_1, n_2)$ is a new estimate of $\lambda(n_1, n_2)$

In Lim and Malik's paper, the calculation of the autocorrelation $R_x(n_1, n_2)$ is limited to the closed analytic form especially for the two-dimensional sinusoids. A generalization of this method and the application to a two-dimensional real data have been discussed by Chen and Young[5]. Even for a small number of missing correlation points, the algorithm can still provide an accurate spectrum. Figure 2a shows the spectrum of two sinusoids (0.1234, 0.3456), (0.3125, 0.200) in white noise, based on a 5x5 correlation data set, at signl to noise ratio of 0.67. With the correlation points (-1, -1) and (1, 1) missing, Fig. 2b shows the resulting spectrum which is nearly identical to Fig. 2a.

## III. Feature Selection and Classfication Method

We use two features to classify the texture images. It is generally recognized that a coarse texture will have a high value of power spectrum near the origin while in a fine texture, the value will be more spread out. Thus, if one wishes to analyze texture coarseness, a set of features that should be useful are the averages of the power-spectrum values taken over a ring-shaped region centered at the origin. In this paper, we consider only the first quadrant of the power spectrum, then

$$\phi_r = \int_0^{\pi/2} \hat{P}_x(r, \theta)d\theta \quad (3)$$

for various values of r, the ring radius.

For the discrete case, this can be written as (for rings between radius $r_1$ and $r_2$):

$$\phi_{r_1 r_2} = \sum_{r_1^2 \leq x^2+y^2 \leq r_2^2} \hat{P}_x(x,y); \quad 0 \leq x,y \leq n-1 \quad (4)$$

Similarly, it is well known also that the angular distribution of power spectrum is sensitive to the directionality of the texture in frequency $\omega$. A texture with many edges or lines in a given direction $\theta$ will have high values of power spectrum around the perpendicular of $\theta + \pi/2$; while in a nondirectional texture the spectrum should also be nondirectional. Thus a good set of features for analyzing the texture directionality should be the averages of the power-spectrum values taken over wedge-shaped regions centered at the origin, i.e.

$$\phi_\theta = \int_0^\infty \hat{P}(r, \theta)dr \quad (5)$$

for the various values of $\theta$, the wedge slope.

For the discrete case, this is (the wedge between $\theta_1$ and $\theta_2$) given by

$$\phi_{\theta_1\theta_2} = \sum_{\substack{\theta_1 \leq \tan^{-1} y/x < \theta_2 \\ 0<x,y \leq n-1}} \hat{P}_x(x, y) \quad (6)$$

The features calculated by (4) and (6) are sensitive to size and orientation respectively, but not to both. In order to obtain the comparable feature sets, we obtain a set of equalized features by taking the average over the intersection area of rings and wedges. These equalized features are also studied in section IV. After the calculation of features, the Fisher discriminant technique is used for classification [6].

## IV. Experimental Results

Because of the computational requirements of the method and the limited memory capacity of the PDP 11/45, all test samples are stored in our DEC 20 system and sent through a communication line to the PDP 11/45 for the spectrum computation. The test samples are the texture images taken from the USC data base. To verify the sensitivity both in coarseness and directionality, we select some textures that contain such informations. The test samples contain six classes of texture (each one has four samples) and are shown in Fig. 3. Each data is a 32x32 array of gray level 0-255. The pictures of class 1 reappear but are two times larger in Fig. 4(a). Figure 4(b) is the corresponding estimated power-spectrum display of the upper left data in each class. The spectra of all classes are different either in radial or angular distribution [7].

The feature sets used in this paper are:

ring: $\phi_{r_1 r_2}$ for $(r_1, r_2)$ = (1,3), (3,6), (6,12), (12,24), (24,48)

wedge: $\phi_{\theta_1 \theta_2}$ for $(\theta_1, \theta_2)$ = (0,15), (15,30), (30,45), (45,60), (60,75), (75,90)

The maximum ring radius used is 48 since it already covers most part of a 64x64 array power spectrum.

A combination feature of ring and wedge has been tested for 30 pairs of feature values. Table I shows part of features which did higher than 19 out of 24 correct, i.e., more than 80% correct. Table II shows the best performing pairs using the same kind of features (ring and ring, wedge and wedge): there are 6 out of 25 pairs which classified correctly higher than 75%. Other pairs' results are concentrated near 12-17, i.e., more than 50% correct recognition. For the pairs that contain the wedge near the edges, the results are very good since the test samples give some directional information. Also for the rings a little farther from the origin, the results are better since it shows a large difference in the spectrum value there.

Equalized features are also tested: we used five rings intersected with three wedges (ring: (1,3), (3,6), (6,12), (12,24), (24,48) and wedge: (0,30), (30,60), (60,90)). 105 pairs of features have been tested. Table III shows the best performing pairs of which the best score, 23 out of 24, is 95% correct. From the results, we can see that the ring feature (24,48) gives very useful classification information indicating that there exists a large textural variation in that region as the texture coarseness plays an important role in the pair.

## V. Discussion

In this paper, we have observed that equalized features did better than unequalized ones for this test samples. It is shown that both the coarseness and the directionality are important factors in texture discrimination. For the consideration of practical use in automatic classification, various kinds of textures must be tested and compared with other methods using the non-spectral features. Another important factor which may influence the results is that if we increase the autocorrelation function and the discrete Fourier transform length while estimating the power spectrum, the accuracy and the resolution will be better. But there is a tradeoff between the accuracy and the computational time. In this paper, these parameters (i.e., autocorrelation function: 7x7, discrete Fourier transform length: 32) are chosen for the real-time processing purpose. Also the locations of the main and second components of frequencies can serve as another important features because they vary among different textures.

Acknowledgement: This work was supported by the ONR Statistics and Probability Program Contract No. N00014-79-C-0494. The programming assistance of Mr. Gia-Kinh Young is gratefully acknowledged.

## References

1. R. Bajcsy and L. Lieberman, "Texture gradient as a depth cue," Computer Graphics and Image Processing, Vol. 5, pp. 52-67, 1976.

2. J. Weszka, D. Dyer and A. Rosenfeld, "A comparative study of texture measures for terrain classification," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-6, pp. 269-285, 1976.

3. K. Laws, "Texture energy measures," in R. Nevatia and A. Sawchuk, "Semiannual Technical Report," University of Southern California, USCIPI Report 910, pp. 18-31, 1979.

4. J.S. Lim and N.A. Malik, "A new algorithm for two-dimensional maximum entropy power spectral estimation," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-29, No. 3, pp. 401-413, June 1981.

5. C.H. Chen and G.K. Young, "On a two-dimensional maximum entropy spectral estimation method for the texture-image analysis," SMU-EE-TR-81-16, Technical Report, SMU, N. Dartmouth, MA, Oct. 1981.

6. R.O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis," Wiley, 1972.

7. C.H. Chen, "Nonlinear Maximum Entropy Spectral Analysis Methods for Signal Recognition," Research Studies Press, England, 1982.

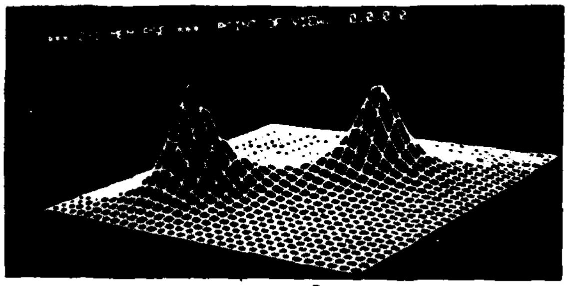


Fig. 2a

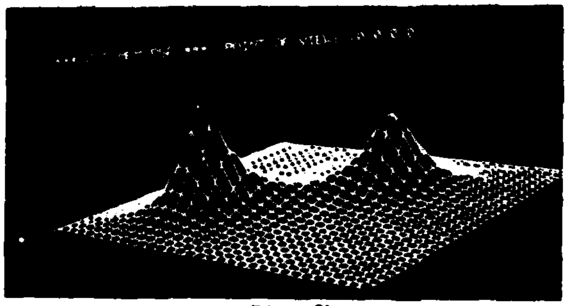


Fig. 2b

$$\text{Initial estimate of } \lambda(n_1,n_2)$$

$$\downarrow$$

$$R_y(n_1,n_2) = F^{-1}\left\{\frac{1}{F[\lambda(n_1,n_2)]}\right\}$$

$$\downarrow$$

$$\text{Correct } R_y(n_1,n_2) \text{ with } R_x(n_1,n_2) \text{ for } (n_1,n_2)\epsilon A$$

$$\downarrow$$

$$\lambda(n_1,n_2) = F^{-1}\left\{\frac{1}{F[R_y(n_1,n_2)]}\right\}$$

$$\downarrow$$

$$\lambda(n_1,n_2) = 0 \text{ for } (n_1,n_2)\epsilon A$$

$$\downarrow$$

$$\hat{P}_x(\omega_1,\omega_2) = F[R_y(n_1,n_2)]$$

Figure 1

Figure 3

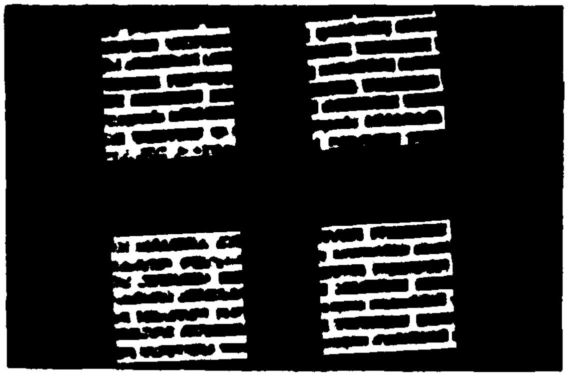


Fig. 4a

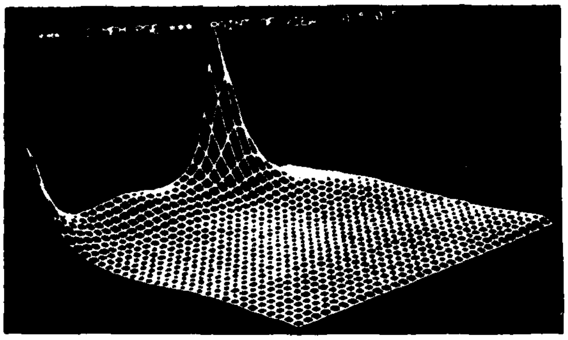


Fig. 4b

| Features: Ring | Features: Wedge | Number correctly classified |
|---|---|---|
| (24,48) | (0,15) | 22 |
| (1,3) | (0,15) | 21 |
| (3,6) | (0,15) | 20 |
| (1,3) | (45,60) | 19 |
| (12,24) | (0,15) | 19 |
| (3,6) | (30,45) | 19 |
| (3,6) | (75,90) | 19 |

Table I: Best performing pairs using the combination feature of ring and wedge for those with more than 80% correct classification.

| Features | | Number correctly classified |
|---|---|---|
| **Ring** | **Ring** | |
| (6,12) | (24,48) | 20 |
| (6,12) | (12,24) | 19 |
| **Wedge** | **Wedge** | |
| (30,45) | (75,90) | 20 |
| (15,30) | (60,75) | 18 |
| (30,45) | (60,75) | 18 |
| (45,60) | (60,75) | 18 |

Table II: Best performing pairs using same kind of features, for those with more than 75% correct classification.

| Features: Ring ∩ | Wedge | Ring ∩ | Wedge | Number correct classification |
|---|---|---|---|---|
| (3,6) | (0,30) | (24,48) | (0,30) | 23 |
| (12,24) | (60,90) | (24,48) | (0,30) | 23 |
| (24,48) | (0,30) | (24,48) | (30,60) | 22 |
| (1,3) | (0,30) | (24,48) | (0,30) | 22 |
| (1,3) | (30,60) | (24,48) | (0,30) | 22 |
| (1,3) | (60,90) | (24,48) | (0,30) | 22 |
| (3,6) | (30,60) | (24,48) | (0,30) | 21 |
| (3,6) | (60,90) | (24,48) | (0,30) | 21 |
| (6,12) | (0,30) | (24,48) | (0,30) | 21 |
| (12,24) | (30,60) | (24,48) | (0,30) | 21 |
| (12,24) | (0,30) | (12,24) | (60,90) | 20 |
| (1,3) | (30,60) | (6,12) | (0,30) | 20 |
| (1,3) | (60,90) | (6,12) | (0,30) | 20 |
| (12,24) | (0,30) | (24,48) | (0,30) | 20 |
| (1,3) | (0,30) | (6,12) | (0,30) | 19 |
| (3,6) | (0,30) | (6,12) | (0,30) | 19 |
| (3,6) | (30,60) | (6,12) | (0,30) | 19 |
| (6,12) | (0,30) | (12,24) | (60,90) | 19 |

Table III: Best performing pairs using equalized features for those with more than 80% correct classification.

# ON A SPATIAL CLUSTERING ALGORITHM FOR IMAGE ANALYSIS

Appendix C

C. H. Chen
Electrical Engineering Department
Southeastern Massachusetts University
North Dartmouth, MA 02747 U.S.A.

## Abstract

A computationally efficient spatial clustering algorithm is presented for image segmentation and compression. The algorithm can automatically determine the minimum number of clusters and can also work on a specified number of clusters. Examples are given on the processing of Seasat images using the algorithm.

## Introduction

Besides the use of acoustic sensors, remotely sensed images can provide essential information on the object extraction and tracking in the ocean environment. Seasat SAR images are a good example. Many automatic image analysis algorithms have been developed. Such algorithms are generally application dependent. For remote sensing images, cluster analysis is important as it reveals the structure of the data from which useful information can be derived. Conventional clustering methods do not preserve the spatial relations in a image. Spatial clustering for image analysis has been considered [1][2]. However feature extraction was not taken into account. Furthermore the computation involved is quite extensive. A more efficient spatial clustering algorithm is developed for minicomputer processing, that employs properly selected features. The clustered image shows various regions (segments) from which desired objects may be extracted. Furthermore considerable image data compression is accomplished with essentially no loss of information. Examples are based exclusively on Seasat images dealing with the ocean environment.

## Algorithm for Spatial Clustering

The algorithm proceeds as follows:

(1) Form a feature set, for each pixel, consisting of local mean and gradient. Other features may also be used.

(2) For each 2x2 subarea, measure the mean vector and dispersion.

(3) Determine the critical dispersion, and calculate the merging distance d.

(4) Merge adjacent subareas with distance less than d to form subregions. Calculate the mean vectors of subregions.

(5) Group these mean vectors into clusters using K-mean algorithm which converges to several cluster centers representing the mean vectors of regions.

For a given inter-region threshold distance, the algorithm can automatically adjust to an appropriate number of clusters. If the number of clusters is specified as in conventional cluster analysis, the algorithm will provide the specified number of clusters.

The image speckles in synthetic aperture are multiplicative in nature. Without removing such noise, the cluster results may still be very noisy. A simple preprocessing method is to use the Sigma filter suggested by J.S. Lee [3]. For each 5x5 (or 3x3) subarea the average gray level value is compared with the three-standard deviation ($3\sigma$) of the normalized image histogram (first order probability density). If the value is within $3\sigma$ then the center pixel is replaced by the average value. If the value exceeds $3\sigma$, then it is an indication of edges or object boundary and the original gray level is retained. The procedure thus provides a compromise between noise filtering and edge preserving and adds only slight amount of computation to the clustering algorithm.

## Computer Results

The original Seasat images are all of 256 gray levels. For convenience with minicomputer processing the digitized pictures are all reduced to 256x256 pixels even though the original images are much larger in size. The cluster algorithm is set such that a maximum of 7 clusters is selected. Figure 1 is for the scene of a ship off Chesapeake Bay. Figure 1a is the original. Figure 1b is the spatially clustered image. Figure 1c has the histograms of original (left) and clustered (right) images. Figure 1d uses the Sigma filter preprocessing with 3x3 subarea while Fig. 1e is the corresponding result with 5x5 subarea. Preprocessing with 5x5 subarea followed by spatial clustering appears to be the best. Figure2 is the scene of Anchorage, Alaska with Fig. 2a for original and Fig. 2b for 5x5 preprocessing followed by spatial clustering. Figure 3 is the scene of Nantucket Shoals with Fig. 3a for original, Fig. 3b for 5x5 median filtering and Fig. 3c for 5x5 preprocessing followed by spatial clustering. Computer results all show that the "natural" clusters of the original images are very much preserved while noises are considerably reduced, and at the same time the contrast is enhanced, with the use of the spatial clustering algorithm.

## Acknowledgement

This work was suported by ONR Contract N00014-79-C-0492. The digital Seasat SAR images were kindly provided by Dr. Jong-Sen Lee of Naval Research Laboratory who also brought to our attention of his work on filtering of multiplicative noises.

## References

1. R.M. Haralick and I. Dinstein, "A spatial clustering procedure for multi-image data", IEEE Trans. on Circuits and Systems, Vol. C-22, pp. 440-450, 1975.
2. Y. Fukada, "Spatial clustering procedures for region analysis", Pattern Recognition, Vol. 12, no. 6, pp. 395-403, 1980.
3. J.S. Lee, "The Sigma filter and its application to speckle smoothing of synthetic aperture radar image", presented at the ONR Signal Processing in the Ocean Environment Workshop held at U.S. Naval Academy, May 1982.

Fig. 1a

Fig. 1b

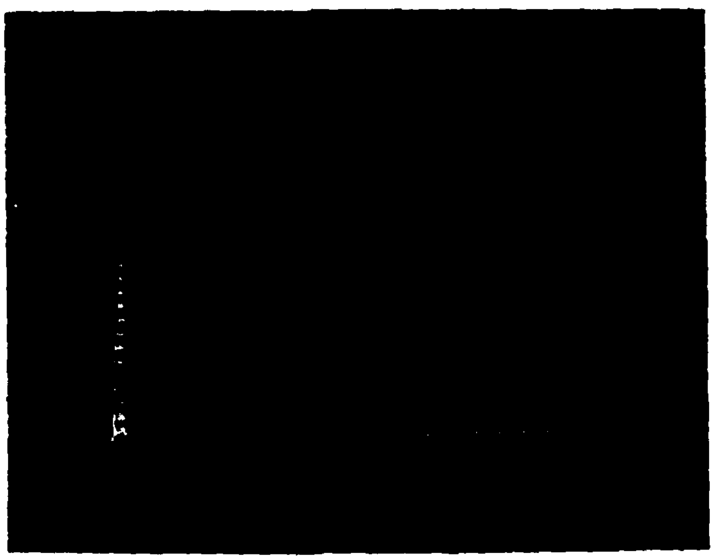


Fig. 1c

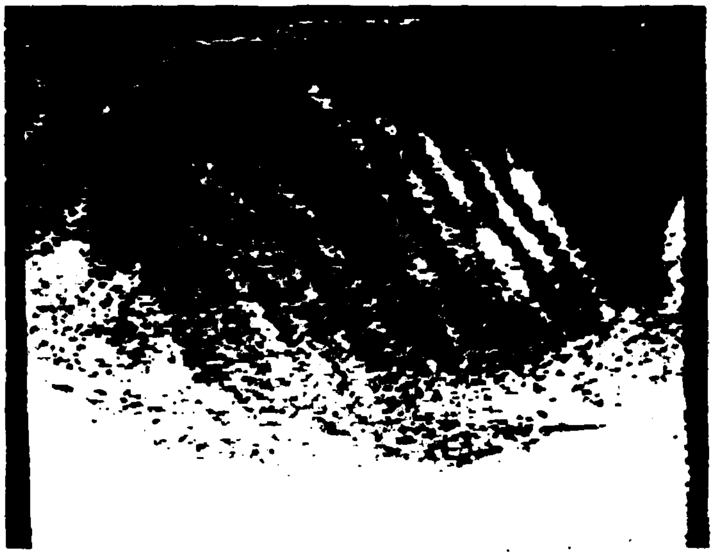


Fig. 1d

Fig. 1e

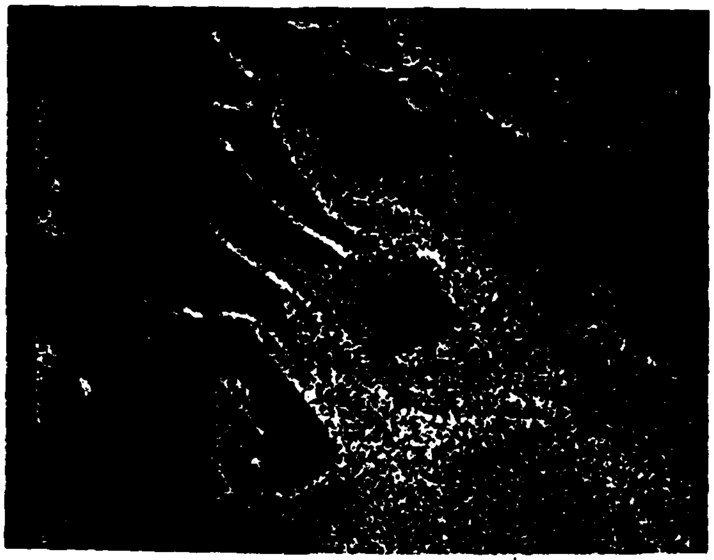


Fig. 2a

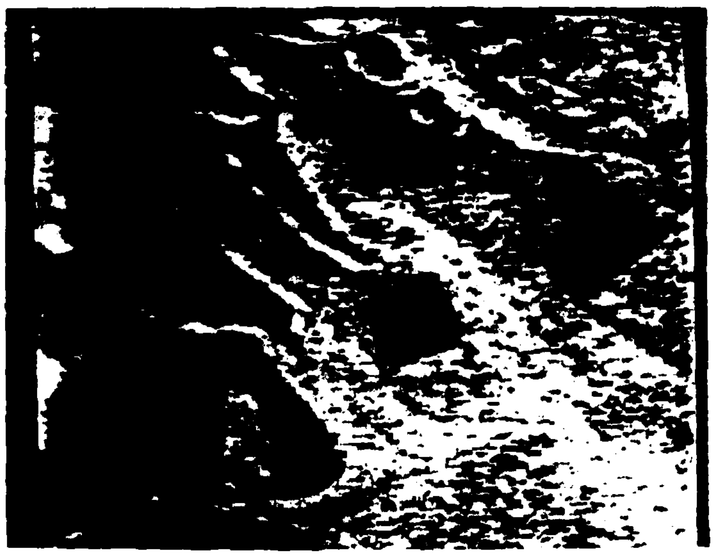


Fig. 2b

Fig. 3a

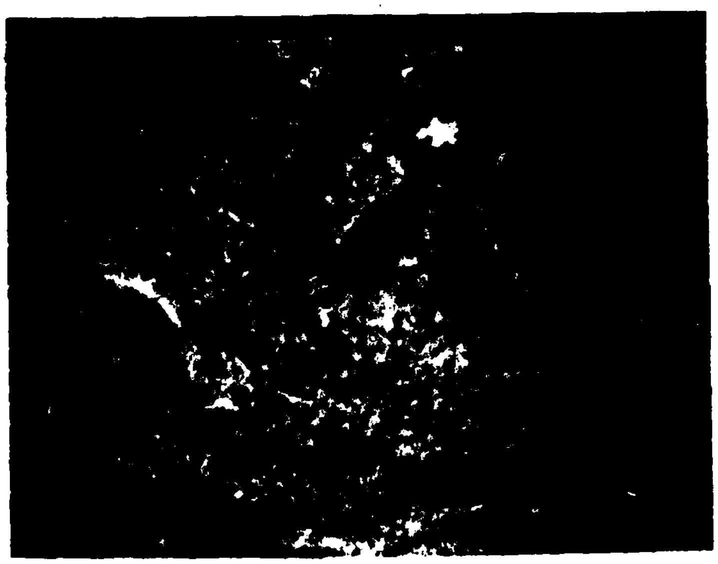


Fig. 3b

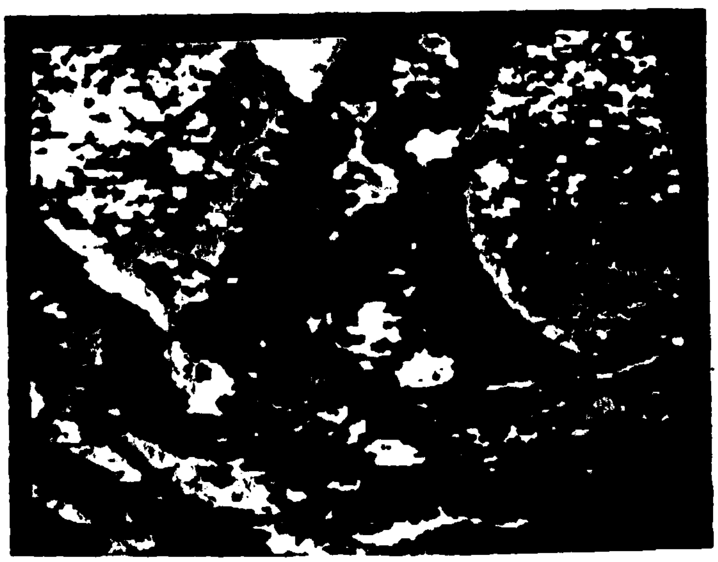


Fig. 3c

APPENDIX PRESPA.FOR

```
C       ******************************************
C       PRESPA.FOR     ( WAS PREA4.FOR )
C       30-MAY-82      ( 300 SECTION : READ DATA FILE FROM TAPE )
C       27-APR-82
C       PROGRAM TO READ A DISK FILE 256X256 PIXELS
C       OUTPUT ITS FEATURE VECTORS, MEAN , GRADIENT
C       OR ORIGINAL INTENSITY , GRADIENT
C       CHOICE 3: READ DATA FROM TAPE
C                 TO FORM NOF1 , NOF2 OUTPUT FILES
C                 FEATURE 1 AND FEATURE 2 RESPECTIVELY
C       NOF1: OUTPUT DATA , MEAN OR ORIGINAL GRAY LEVEL,  COMPONENT
1
C       NOF2: OUTPUT DATA , GRADIENT, COMPONENT 2
C       IF CHOICE 1 OR 2 : READ DISK FILE NOF4
C       NOF4: INPUT DATA
C       ******************************************
        INTEGER F(1024),F2(1024),CHOICE,IMEAN(256),IGRAD(256)
        REAL DMEAN(256),DGRAD(256),WS1(256),WS2(256)
        DATA NOF1,NOF2,NOF4/1,2,4/
1001    FORMAT(' PROGRAM PRESPA.FOR'/
        2        ' PREPROCESSING IMAGE DATA TO FORM FEATURE
VECTORS'/
        3        ' FOR AUTO SPATIAL CLUSTERING'/
        4        ' INPUT:  FTN4.DAT OR TAPE - ORIGINAL GRAY LEVEL'/
        5        ' OUTPUT: FTN1.DAT FEATURE COMPONENT 1'/
        6        '         FTN2.DAT, FEATURE COMPONENT 2'/
        7        ' CHOICES:'/' 1: LOCAL MEAN , LOCAL GRADIENT'/
        8        ' 2: ORIGINAL INTENSITY , LOCAL GRADIENT'/
        9        ' 3: COMPRESS A 2BY 2 SUBIMAGE INTO 1 PIXEL'/
        1        ' 4: READ TAPE DATA (NORGL*NORGP) TO
DISK(NOL*NOP)'/)
1002    FORMAT(I5)
1003    FORMAT(' ENTER INPUT AND OUTPUT FILES SIZE'/
        1        ' NOLIN,NOPIN,NOL,NOP: FORMAT(4I5)')
1004    FORMAT(4I5)
1       CONTINUE
        WRITE(7,1001)
        READ(5,1002)CHOICE
        IF (CHOICE.LE.0.OR.CHOICE.GT.4) GOTO 1
        GOTO (50,50,300,400),CHOICE
50      CONTINUE
        WRITE(7,1003)
        READ(5,1004)NOLIN,NOPIN,NOL,NOP
C       ORIGINAL INTEGER DATA FILE OF AN IMAGE
        DEFINE FILE NOF4(NOLIN,NOPIN,U,INDX4)
        NOL1=NOL-1
        NOP1=NOP-1
C       OUTPUT FILES, NOF1 , NOF2, INTEGER NUMBERS
        DEFINE FILE NOF1(NOL,NOP,U,INDX1)
        DEFINE FILE NOF2(NOL,NOP,U,INDX2)
        GOTO (100,200),CHOICE
100     CONTINUE
```

```
      DO 90 I=1,NOL1
      INDX4=I
      READ(NOF4'INDX4)(F(K),K=1,NOPIN)
      READ(NOF4'INDX4)(F2(K),K=1,NOPIN)
      DO 70 J=1,NOP1
      IMEAN(J)=(F(J)+F(J+1)+F2(J)+F2(J+1))/4
      IGRAD(J)=(IABS(F(J)-F(J+1))+IABS(F2(J)-F(J)))/2
70    CONTINUE
      IMEAN(NOP)=IMEAN(NOP1)
      IGRAD(NOP)=IGRAD(NOP1)
      INDX1=I
      INDX2=I
      WRITE(NOF1'INDX1)(IMEAN(K),K=1,NOP)
      WRITE(NOF2'INDX2)(IGRAD(K),K=1,NOP)
90    CONTINUE
      WRITE(NOF1'INDX1)(IMEAN(K),K=1,NOP)
      WRITE(NOF2'INDX2)(IGRAD(K),K=1,NOP)
      GOTO 900
200   CONTINUE
C     CHOICE 2: SECTION
C     FEATURE: ORIGINAL INTENSITY , LOCAL GRADIENT
      DO 290 I=1,NOL1
      INDX4=I
      READ(NOF4'INDX4)(F(K),K=1,NOPIN)
      READ(NOF4'INDX4)(F2(K),K=1,NOPIN)
      DO 270 J=1,NOP1
      IGRAD(J)=(IABS(F(J)-F(J+1))+IABS(F2(J)-F(J)))/2
270   CONTINUE
      IGRAD(NOP)=IGRAD(NOP1)
      INDX1=I
      INDX2=I
      WRITE(NOF1'INDX1)(F(K),K=1,NOP)
      WRITE(NOF2'INDX2)(IGRAD(K),K=1,NOP)
290   CONTINUE
      WRITE(NOF1'INDX1)(F2(K),K=1,NOP)
      WRITE(NOF2'INDX2)(IGRAD(K),K=1,NOP)
      GOTO 900
300   CONTINUE
C     READ TAPE DATA FILE SECTION
C     NOLI,NOLF,NOPI,NOPF: COVER THE AREA INTERESTED
C     NOL, NOP:  SIZE OF THE OUTPUT DATA IN FEATURE SPACE
C             EACH RECORD REPRESENTS 2 BY 2 ORIGINAL PIXELS
      WRITE(7,1031)
1031  FORMAT(' ENTER ORIGINAL TAPE FILE SIZE AND WHICH FILE IN
TAPE'/
      1        ' NORGL,NORGP,NTH ( FORMAT(3I6) ):')
      READ(5,1032)NORGL,NORGP,NTH
1032  FORMAT(3I6)
      WRITE(7,1033)
1033  FORMAT(' WHICH PART OF IMAGE TO BE PROCESSED?'/
      1        ' NOLI,NOLF,NOPI,NOPF: ( FORMAT(4I6) )')
      READ(5,1034)NOLI,NOLF,NOPI,NOPF
1034  FORMAT(4I6)
      NOL=(NOLF-NOLI+1)/2
      NOP=(NOPF-NOPI+1)/2
```

```
      WRITE(7,1035)NOLI,NOLF,NOPI,NOPF,NOL,NOP
1035  FORMAT(' CHECK: NOLI,NOLF,NOPI,NOPF,NOL,NOP'/
     1      6I6
     2      /' NOL X NOP WILL BE THE OUTPUT SIZE')
      DEFINE FILE NOF1(NOL,NOP,U,INDX1)
      DEFINE FILE NOF2(NOL,NOP,U,INDX2)
      REWIND NOF1
      REWIND NOF2
C     SKIP THE PART NOT TO BE PROCESSED
      NSKIP=NOLI-1
C     NOLI IS THE FIRST LINE TO BE PROCESSED
      IF (NSKIP.LT.1) GOTO 312
      DO 310 I=1,NSKIP
310   CALL READUN(F,1,NTH)
312   CONTINUE
      DO 350 I=1,NOL
      DO 315 J=1,NOP
      IMEAN(J)=0
      IGRAD(J)=0
315   CONTINUE
      CALL READUN(F,1,NTH)
      CALL READUN(F2,1,NTH)
      DO 320 J=1,NOP
      IMEAN(J)=(F(NOPI+J+J-2)+F(NOPI+J+J-1)+F2(NOPI+J+J-2)+
     1      F2(NOPI+J+J-1))/4
      IGRAD(J)=(IABS(F(NOPI+J+J-2)-F(NOPI+J+J-1))+
     1      IABS(F(NOPI+J+J-2)-F2(NOPI+J+J-2)))/2
320   CONTINUE
      INDX1=I
      INDX2=I
      WRITE(NOF1'INDX1)(IMEAN(K),K=1,NOP)
      WRITE(NOF2'INDX2)(IGRAD(K),K=1,NOP)
350   CONTINUE
      GOTO 900
400   CONTINUE
C     READ TAPE DATA FILE TO FORM A COMPRESSED DISK DATA FILE
C     INPUT SIZE : NORGL BY NORGP PIXELS
C     OUTPUT SIZE: NOL BY NOP
      WRITE(7,1041)
1041  FORMAT(' ENTER NORGL,NORGP,NOL,NOP,NOF,NTH'/
     1      '    E.G., 256,256,64,64,?,?')
      READ(5,1042)NORGL,NORGP,NOL,NOP,NOF,NTH
1042  FORMAT(6I6)
      DEFINE FILE NOF(NOL,NOP,U,INDEX)
      ITERL=NORGL/NOL
      ITERP=NORGP/NOP
      DO 450 I=1,NOL
      DO 420 J=1,ITERL
420   CALL READUN(F,1,NTH)
      WRITE(NOF'INDEX)(F(ITERP*K),K=1,NOP)
450   CONTINUE
      GOTO 900
900   CONTINUE
      CALL EXIT
      END
```

APPENDIX SPABW.FOR

```
C     -------------------------------------------------------------------
----
C     9A123456789B123456789C123456789D123456789E123456789F12345678
9G12
C     FILE NAME: SPABW.FOR
C     AUTO SPATIAL CLUSTERING FOR BLACK/WHITE IMAGE DATA
C
C     REFERENCE FUKADA," SPATIAL CLUSTERING PROCEDURESFOR REGION
C             ANALYSIS",PATTERN RECOGNITION,12,395-403,(1980).
C
C
C     INPUTS: NOF1, FTN1.DAT, FEATURE 1
C             NOF2, FTN2.DAT, FEATURE 2
C     OUTPUTS:NOFC, FTN12.DAT, CLUSTERING RESULT FOR COLOR
DISPLAY
C             NOFB, FTN11.DAT, CLUSTERING RESULT FOR BLACK/WHITE
C                               DISPLAY
C             NOFF, FTN15.DAT, SOME PARAMETERS DURONG PROCESSING
C     SIZE OF IMAGE IS RESTICTED TO 256 BY 256 IN FEATURE SPACE
C     -------------------------------------------------------------------
----
      LOGICAL*1 DDMMY(9)
      REAL PREV(60,2),PRENO(60)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
      COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
      COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
      DATA NOF1,NOF2,NOFB,NOFC,NOFF/1,2,11,12,15/
1701  FORMAT(' RUNNING PROGRAM SPABW.FOR AUTO SPATIAL
CLUSTERING'/
      1       ' INPUTS , OUTPUTS: NOF1,NOF2,NOFB,NOFC,NOFF'/
      6       ' FOR CHECK:',5I5/
      2       ' NOFC , NOFB STORE TEMPORARY DATA DURING
PROCESSING'/
      3       ' OUTPUTS: NOFC(REWRITTEN) 7 OR LESS COLORS'/
      4       '          NOFB(REWRITTEN) BLACK AND WHITE
DISPLAY'/
      5       '          NOFF, INFORMATIONS DURING PROCESSING')
1515  FORMAT('  MERGE ITERATION',I5)
1520  FORMAT(' THE ',I5,'-TH ITERATION REACHES MAXIMUM NO.
CLUSTERS'/
      1       ' NO. OF CLUSTERS:',I5)
1511  FORMAT(' KERNEL CANDIDATE VECTORS FOR',I4,' STARTING ',
      1       'CLUSTER CENTERS'/)
1512  FORMAT(' ENTER IMAGE DATA FILE SIZE ( FEATURE SPACE )'/
      1       ' NOL, NOP: FORMAT(2I5)')
1513  FORMAT(2I5)
1514  FORMAT(' CHECK:NOL,NOP,NOL2,NOP2,NOLH,NOPH'/6I5)
1516  FORMAT(' ENTER OPTIONS: (IOP(K),K=1,10)'/
      1       '  IOP(1): CONTROLS PRINTER, 1: MEANS , TRACE
MATRIX'/
      2       '  IOP(2): K-MEANS ALGORITHM DETAILS ON SCREEN'/
      3       '  IOP(3): MERGE DETAILS, LABEL(2,K) ARRAY'/
```

```
     4        '  IOP(4): NSTEP, NO. OF STEPS'/
     5        '  IOP(5): 1, SKIP K-MEAN ITERATION'/)
1517  FORMAT(10I6)
1518  FORMAT(' TODAY IS ',9A1)
      REWIND NOFF
      WRITE(7,1501)
      WRITE(NOFF,1501)
      CALL DATE(DDMMY)
      WRITE(NOFF,1518)(DDMMY(K),K=1,9)
      WRITE(7,1518)   (DDMMY(K),K=1,9)
      WRITE(NOFF,1701)NOF1,NOF2,NOFB,NOFC,NOFF
      WRITE(7,1701)NOF1,NOF2,NOFB,NOFC,NOFF
      WRITE(7,1512)
      WRITE(NOFF,1512)
      READ(5,1513)NOL,NOP
      WRITE(7,1516)
      WRITE(NOFF,1516)
      READ(5,1517)(IOP(K),K=1,10)
      WRITE(NOFF,1517)(IOP(K),K=1,10)
      WRITE(7,1517)(IOP(K),K=1,10)
      NOL2=NOL+NOL
      NOP2=NOP+NOP
      NOLH=NOL/2
      NOPH=NOP/2
      WRITE(NOFF,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
      WRITE(7,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
      DEFINE FILE NOF1(NOL,NOP,U,INDX1)
      DEFINE FILE NOF2(NOL,NOP,U,INDX2)
      DEFINE FILE NOFB(NOL,NOP,U,INDXB)
      DEFINE FILE NOFC(NOL,NOP,U,INDXC)
C     NOFF=FTN15.DAT UNFORMATTED
      REWIND NOF1
      REWIND NOF2
      REWIND NOFB
      REWIND NOFC
1501  FORMAT(' THIS IS THE LOG FILE OF EXECUTING SPABW.FOR')
C     ------------------------------------------
C     CALCULATE MEANS OF FEATURE VECTORS OF 2 BY 2 SUBIMAGE
C     STORE IN NOFB: FIRST HALF -- FEATURE 1 MEANS OF 128X128
C                    SECOND HALF -- FEATURE 2 MEANS
C                    128 X 128 SUBIMAGES EACH
C     IN NOFC: FIRST AND SECOND HALF ARE THE SAME, TRACES
      CALL DISPER
C     ------------------------------------------
C     FIND MAX , MIN OF TRACE MATRIX
      CALL MAXMIN(DMAX,DMIN)
C     ------------------------------------------
C     MERGING SECTION
      NSTEP=IOP(4)
      STEP=(DMAX-DMIN)/FLOAT(NSTEP)
      WRITE(7,1522)DMAX,DMIN,STEP
      WRITE(NOFF,1522)DMAX,DMIN,STEP
1522  FORMAT(' DMAX=',E20.8,' DMIN=',E20.8,' STEP=',E20.8)
      IPREV=0
C     ITERATIONS TO FIND MAXIMUM NO. OF CLUSTERS
```

```
      DO 300 I=1,NSTEP
      IM=I
      WRITE(NOFF,1515)IM
      CALL MERGE(IM,STEP,DMIN,NCLSR)
C     ACCEPTED NO. OF CLUSTERS: 7 OR LESS
      IF (IPREV.LE.7.AND.IPREV.GT.1) GOTO 333
      IPREV=NCLSR
C     SAVE CURRENT NUMBER OF CLUSTERS AND KERNEL VECTORS
      DO 200 J=1,IPREV
      PRENO(J)=CNO(J)
      PREV(J,1)=CKV(J,1)
      PREV(J,2)=CKV(J,2)
200   CONTINUE
300   CONTINUE
333   NI=IM-1
      WRITE(NOFF,1520)NI,IPREV
      DO 350 J=1,IPREV
      CNO(J)=PRENO(J)
      CKV(J,1)=PREV(J,1)
      CKV(J,2)=PREV(J,2)
350   CONTINUE
      WRITE(NOFF,1561)
1561  FORMAT(' MERGE ENDED WITH MAXIMUM NO. CLUSTERS')
      WRITE(NOFF,1562)((CKV(N,L),L=1,2),N=1,IPREV)
1562  FORMAT(' BEFORE SORTING'/
     1         ' KERNEL CANDIDATE VECTORS'/
     2         30((5X,2E20.8)/))
C     SORT THE CANDIDATE VECTORS
      NC=IPREV
C     ---------------------------------------------------
C     SORT THE CANDIDATE KERNEL VECTORS
      CALL SORT(NC)
      WRITE(NOFF,1563)((CKV(N,L),L=1,2),N=1,NC)
1563  FORMAT(' SORTED KERNEL CANDIDATE VECTORS'/
     1         30((5X,2E20.8)/))
      IF (NC.GT.7) NC=7
C     FOR THE PURPOSE OF AED-512 PSEUDO COLOR DISPLAY
      WRITE(NOFF,1511)NC
C     IF (IOP(5).EQ.1) SKIP THE K-MEAN ITERATIONS
C     DIRECTLY USE MERGING RESULT CADIDATE KERNEL VECTORS
C     TO CLASSIFY THE IMAGE
      WRITE(7,1568)IOP(5)
1568  FORMAT(' IOP(5)=',I5)
      IF (IOP(5).NE.1) GOTO 700
      WRITE(7,1570)
1570  FORMAT(' SKIP K-MEAN ITERATION')
      DO 650 N=1,NC
      DO 640 L=1,2
      FKV(N,L)=CKV(N,L)
640   CONTINUE
650   CONTINUE
      GOTO 800
700   CONTINUE
      WRITE(7,1580)
1580  FORMAT(' CALLING KERVEC: K-MEAN ITERATION')
```

APPENDIX SPABW.FOR

```
C     ---------------------------------------------------
C     ITERATIONS TO FIND MORE ACCURATE KERNEL VECTORS
C     CALLED FINAL KERNEL VECTORS
      CALL KERVEC(NC,KK,DD)
      WRITE(NOFF,1500)KK,DD
1500  FORMAT(1X,'CLUSTERING REPEATS',1X,I3,1X,'TIMES'/1X,
     1'THE FINAL WITHIN-CLASS DISTANCE IS',1X,E20.8/)
800   CONTINUE
C     ---------------------------------------------------
C     CLASSIFICATION SECTION
C     OUTPUTS: NOFC, COLOR DISPLAY RESULT
C              NOFD, BLACK/WHITE DISPLAY RESULT
      CALL CLASS(NC)
      WRITE(NOFF,1523)
1523  FORMAT(10X,'!!! COMPLETE EXECUTION OF PROGRAM SPABW !!!')
      CALL EXIT
      END
C
C
C     ----------- SUBPROGRAMS ----------------
C
C     ---------------------------------------------------
C     SUBROUTINE TO CALCULATE TRACE MATRICES OF FEATURE MATRICES
C     STORED IN NOFD
      SUBROUTINE DISPER
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
      COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
      REWIND NOF1
      REWIND NOF2
      REWIND NOFB
      REWIND NOFC
C     PROCESS THROUGH ROWS OF DATA MATRIX
      DO 100 I=1,NOLH
      I2=I+I-1
      INDX1=I2
      INDX2=I2
      DO 40 JJ=1,2
C     READ 2 LINES OF EACH FILE
      READ(NOF1'INDX1)(IA(K),K=1,NOP)
      DO 10 J=1,NOP
10    A(J,JJ)=FLOAT(IA(J))
      READ(NOF2'INDX2)(IB(K),K=1,NOP)
      DO 20 J=1,NOP
20    B(J,JJ)=FLOAT(IB(J))
40    CONTINUE
C     CALCULATION THROUGH EACH SUBIMAGE
      DO 80 K=1,NOPH
      K1=K+K-1
      K2=K1+1
      S1=0.
      S2=0.
      DO 62 M=K1,K2
      DO 60 L=1,2
```

```
      S1=S1+A(M,L)
      S2=S2+B(M,L)
60    CONTINUE
62    CONTINUE
      ABM(2,K,1)=S1*0.25
      ABM(2,K,2)=S2*0.25
      S1=0.
      S2=0.
      DO 72 M=K1,K2
      DO 70 L=1,2
      S1=S1+(A(M,L)-ABM(2,K,1))**2
      S2=S2+(B(M,L)-ABM(2,K,2))**2
70    CONTINUE
72    CONTINUE
      TRACE(K)=(S1+S2)*0.25
80    CONTINUE
      INDXB=I
      IF (IOP(1).EQ.1) WRITE(6,1001)(ABM(2,K,1),K=1,32)
      WRITE(NOFB'INDXB)(ABM(2,K,1),K=1,NOPH)
      INDXB=I+NOLH
      IF (IOP(1).EQ.1) WRITE(6,1002)(ABM(2,K,2),K=1,32)
      WRITE(NOFB'INDXB)(ABM(2,K,2),K=1,NOPH)
      INDXC=I
      WRITE(NOFC'INDXC)(TRACE(K),K=1,NOPH)
      INDXC=I+NOLH
      IF (IOP(1).EQ.1) WRITE(6,1004)(TRACE(K),K=1,32)
      WRITE(NOFC'INDXC)(TRACE(K),K=1,NOPH)
100   CONTINUE
1001  FORMAT(' AM',32F4.0)
1002  FORMAT(' BM',32F4.0)
1004  FORMAT(' TR',32F4.0)
      RETURN
      END
C     ------------------------------------------------------
C     READ TRACE MATRIX TO FIND MAX , MIN
C
      SUBROUTINE MAXMIN(DMAX,DMIN)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
      COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
      REWIND NOFC
      INDXC=1
      READ(NOFC'INDXC)(TRACE(K),K=1,NOPH)
      DMAX=TRACE(1)
      DMIN=TRACE(1)
      DO 10 J=2,NOPH
      IF (TRACE(J).LT.DMIN) DMIN=TRACE(J)
      IF (TRACE(J).GT.DMAX) DMAX=TRACE(J)
10    CONTINUE
      DO 100 I=2,NOLH
      INDXC=I
      READ(NOFC'INDXC)(TRACE(K),K=1,NOPH)
      DO 30 J=1,NOPH
      IF (TRACE(J).LT.DMIN) DMIN=TRACE(J)
```

```
      IF (TRACE(J).GT.DMAX) DMAX=TRACE(J)
30    CONTINUE
100   CONTINUE
      WRITE(7,1001)DMAX,DMIN
1001  FORMAT(/' DMAX=',F12.4,' DMIN=',F12.4)
      RETURN
      END
C
C     -------------------------------------------
C     MERGE AND DECIDE KERNEL CANDIDATE VECTORS
C
      SUBROUTINE MERGE(IMRGE,DSTEP,DMIN,LLBS)
      REAL FIRST(60)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
      COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
      COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1502  FORMAT(30(/' CNO(N):',F12.1))
1503  FORMAT(' *FMIN, DMRGE:',2E20.8)
1504  FORMAT(/' IMRGE:',I5)
1505  FORMAT(' FIRST SUBIMAGE:')
1506  FORMAT(' JOIN CLUSTER NL:')
1507  FORMAT(' NEW CLUSTER:')
1508  FORMAT(' LLBS.GE.60')
1532  FORMAT(15X,'THETA',7X,' SIGMA-SQUARE',3X,'
MERGING-DISTANCE:'/
      1        3E20.8)
1533  FORMAT(' K,LLBS,TRACE(K),THETA,DMRGE:'/2I5,3E20.8)
      REWIND NOFB
      REWIND NOFC
C     FOR EACH ITERATION: ZERO OUT THE VARIABLES
      DO 20 J=1,60
      CNO(J)=0.
      DO 10 L=1,2
      CKV(J,L)=0.
10    CONTINUE
20    CONTINUE
      LLBS=0
C     DMIN: SIGMA-SQUARE
C     THETA:  SOME THETA
C     DMRGE:    MERGING DISTANCE
      THETA=DMIN+DSTEP*FLOAT(IMRGE)
      DMRGE=SQRT(4./3.*(THETA-DMIN))
      WRITE(NOFF,1532)THETA,DMIN,DMRGE
      WRITE(7,1504)IMRGE
C     GO THROUGH SUBIMAGES AND LABEL THEM WITH CLUSTERS
      DO 300 J=1,NOLH
      IF (J.GT.1) GOTO 35
C     J=1 : CASE OF FIRST LINE OF SUBIMAGES
      DO 30 K=1,NOPH
      LABEL(2,K)=0
      DO 30 L=1,2
30    ABM(1,K,L)=0.
```

```
      GOTO 45
35    CONTINUE
C     GET PREVIOUS LINE OF SUBIMAGES IN RGBM ARRAY
      DO 40 K=1,NOPH
      DO 40 L=1,2
40    ABM(1,K,L)=ABM(2,K,L)
45    CONTINUE
      INDXB=J
      READ(NOFB'INDXB)(ABM(2,K,1),K=1,NOPH)
      INDXB=J+NOLH
      READ(NOFB'INDXB)(ABM(2,K,2),K=1,NOPH)
C     INTIAL LABEL FOR EACH SUBIMAGE
      DO 50 K=1,NOPH
      LABEL(1,K)=LABEL(2,K)
      LABEL(2,K)=0
50    CONTINUE
      INDXC=J
      READ(NOFC'INDXC)(TRACE(K),K=1,NOPH)
C     GO THROUGH IMAGES ONE BY ONE
      DO 201 K=1,NOPH
      IF (IOP(1).EQ.9) WRITE(7,1533)K,LLBS,TRACE(K),THETA,DMRGE
      IF (LLBS.GE.60) WRITE(7,1508)
      IF (LLBS.GE.60) GOTO 900
C     CHECK IF THE TRACE OF CURRENT SUBIMAGE > THETA
      IF (TRACE(K).GT.THETA) GOTO 200
C     SKIP
      IF (J.GT.1) GOTO 52
C     J=1: FIRST LINE OF SUBIMAGES
C     THE FIRST LINE SECTION: CONSIDERING THE NEIGHBOR
      M1=2
      M2=2
      K1=K-1
      K2=K
      GOTO 54
52    CONTINUE
C     NOT THE FIRST LINE; SO PREVIOUS LINE EXISTS
      M1=1
      M2=2
      K1=K
      K2=K
54    CONTINUE
C     CHECK IF FIRST SUBIMAGE OR NOT
      IF (LLBS.EQ.0) GOTO 90
      IF (LABEL(M1,K1).EQ.0) GOTO 55
C     POTENTIAL NEIGHBOR NOT LABELLED, DIRECTLY CHECK CLUSTERS
C     LABEL(M1,K1) NEIGHBOR HAS BEEN LABELLED
C     AND SPATIAL CLUSTERING SHOULD BE APPLIED
      DIFF=0.
      DO 62 L=1,2
62    DIFF=DIFF+(ABM(M1,K1,L)-ABM(M2,K2,L))**2
      DIFF=SQRT(DIFF)
      IF (DIFF.GT.DMRGE) GOTO 55
C     WITHIN MERGING DISTANCE ?
      LABEL(M2,K2)=LABEL(M1,K1)
      NL=LABEL(M1,K1)
```

```
      LABEL(M2,K2)=NL
      CNO(NL)=CNO(NL)+1.
      DO 64 L=1,2
64    CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+ABM(M2,K2,L))/CNO(NL)
      GOTO 200
55    CONTINUE
      DO 58 N=1,LLBS
      DIFF=0.
      DO 56 L=1,2
      DIFF=DIFF+(CKV(N,L)-ABM(2,K2,L))**2
56    CONTINUE
      FIRST(N)=SQRT(DIFF)
58    CONTINUE
      CALL DISMIN(FIRST,FMIN,NL,LLBS)
      IF (FMIN.GT.DMRGE) GOTO 90
      IF (IOP(1).EQ.9) WRITE(7,1503)FMIN,DMRGE
C     LABEL CURRENT SUBIMAGE WITH CLOSEST CENTER
      LABEL(M2,K2)=NL
C     UPDATE NO. OF SUBIMAGES OF CURRENT CLUSTER
      CNO(NL)=CNO(NL)+1.
C     UPDATE MEAN VECTOR OF THIS CLUSTER
      DO 60 L=1,2
      CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+ABM(2,K2,L))/CNO(NL)
60    CONTINUE
      IF (IOP(1).EQ.9) WRITE(7,1506)
      GOTO 200
C     NEW CLUSTER SECTION
90    CONTINUE
      LLBS=LLBS+1
C     UPDATE   OF SUBIMAGES OF THIS CLUSTER
      CNO(LLBS)=CNO(LLBS)+1.
C     UPDATE NEW CLUSTER VECTOR
      DO 92 L=1,2
92    CKV(LLBS,L)=ABM(M2,K,L)
200   CONTINUE
201   CONTINUE
250   CONTINUE
C     CHECK CURRENT LINE'S LABELS
      IF (IOP(3).EQ.1) WRITE(7,1545)(LABEL(2,K),K=1,32)
300   CONTINUE
900   CONTINUE
      IF (IOP(1).EQ.9) WRITE(7,1502)(CNO(N),N=1,LLBS)
      IF (IOP(1).EQ.9) WRITE(7,1501)IMRGE,LLBS
      WRITE(NOFF,1501)IMRGE,LLBS
1545  FORMAT(' LABEL',32I2)
1501  FORMAT(/' MERGE ITERATION:',I5,' END WITH LLBS: ',I5/)
      RETURN
      END
C
C     ------------------------------------------------------------
C     SORTING THE KERNEL VECTORS
C
      SUBROUTINE SORT(NCLRS)
      REAL TEMP(2)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
```

```
      COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
      DO 30 I=2,NCLRS
      I2=NCLRS+1-I
      DO 20 J=1,I2
      IF (CKV(J+1,1).GE.CKV(J,1)) GOTO 20
      DO 10 L=1,2
      TEMP(L)=CKV(J+1,L)
      CKV(J+1,L)=CKV(J,L)
      CKV(J,L)=TEMP(L)
10    CONTINUE
20    CONTINUE
30    CONTINUE
      WRITE(NOFF,1533)
1533  FORMAT(/' SORTING CANDIDATE KERNEL VECTORS'/)
      RETURN
      END
C
C     ----------------------------------------------
C     TO FIND FINAL KERNEL VECTORS
C     LIMIT TO 10 ITERATIONS
C
      SUBROUTINE KERVEC(NC,KK,DIS)
C     DIST ARRAY STORES THE TOTAL DISTANCES OF ITERATIONS
C     C ARRAY STORES NUMBER OF PIXELS FOR EACH CLUSTER
C     D ARRAY STORES TEMPORARY DISTANCES TO CLUSTER CENTERS
C        FOR CURRENT PIXEL BEING PROCESSED
      REAL DIST(10),FIRST(60)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
      COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
      COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1020  FORMAT(' IN KERVEC, KK:',I5)
C     FINAL KERNELVECTORS SAVED IN FKV ARRAY
C     FNO STORE NO. OF PIXELS IN EACH CLUSTER
C     K-MEANS ALGORITHM
C     10-JUN-82 CORRECT IMPLEMENTATION
C     REFERENCE: TOU AND GONZALEZ," PATTERN RECOGNITION
C      PRINCIPLES", PP.94-97,(1974).
      DO 12 N=1,NC
      FNO(N)=CNO(N)
      CNO(N)=0.
      DO 10 L=1,2
      FKV(N,L)=CKV(N,L)
      CKV(N,L)=0.
10    CONTINUE
12    CONTINUE
C     NO. OF ITERATIONS LIMIT TO 10
      DO 500 KK=1,10
      WRITE(7,   1020)KK
      WRITE(NOFF,1020)KK
      DO 15 N=1,NC
15    FNO(N)=0.
      REWIND NOF1
```

```
      REWIND NOF2
      REWIND NOFB
C     FOR EACH ITERATION:
C     REWIND THE FEATURES FILES: A , B COMPONENTS
C     REWIND THE TEMPORARY CLASSIFIED RESULT FILE, NOFB
C     CLASSIFYING STANDARDS IN FKV ARRAY
C     AT THE SAME TIME, COLLECTING THE NEW CENTERS IN CKV ARRAY
C     I.E., UPDATING THE KERNEL VECTORS BY CURRENT CLUSTERING
C     ICOLOR ARRAY STORES CLASSIFIED RESULT OF CURRENT LINE
      DO 200 I=1,NOL
      INDX1=I
      READ(NOF1'INDX1)(IA(K),K=1,NOP)
      INDX2=I
      READ(NOF2'INDX2)(IB(K),K=1,NOP)
      DO 20 J=1,NOP
      AB(J,1)=FLOAT(IA(J))
      AB(J,2)=FLOAT(IB(J))
20    CONTINUE
C     GO THROUGH PIXELS TO LABEL THEM WITH CLUSTERS
      DO 100 J=1,NOP
      DO 40 N=1,NC
      SUM=0.
      DO 30 L=1,2
30    SUM=SUM+(AB(J,L)-FKV(N,L))**2
40    FIRST(N)=SQRT(SUM)
      CALL DISMIN(FIRST,FMIN,NN,NC)
      ICOLOR(J)=NN
      CNO(NN)=CNO(NN)+1.
C     CURRENT PIXEL WAS FOUND CLOSER TO NN-TH CLUSTER
C     THE FEATURES SHOULD BE INCLUDED TO UPDATE THE NN-TH
C     KERNEL VECTOR
      DO 70 L=1,2
70    CKV(NN,L)=(CKV(NN,L)*(CNO(NN)-1.)+AB(J,L))/CNO(NN)
100   CONTINUE
      INDXB=I
      IF (IOP(2).EQ.1) WRITE(7,1505)(ICOLOR(K),K=1,64)
      WRITE(NOFB'INDXB)(ICOLOR(K),K=1,NOP)
200   CONTINUE
C     CURRENT IN CKV; STORE IN FKV TO BE USED TO CLASSIFY
C     IN ROUTINE DISTAN, AND WILL GIVE TOTAL DISTANCE
      DO 260 N=1,NC
      FNO(N)=CNO(N)
      CNO(N)=0.
      DO 250 L=1,2
      FKV(N,L)=CKV(N,L)
      CKV(N,L)=0.
250   CONTINUE
260   CONTINUE
      WRITE(NOFF,1503)((FKV(N,L),L=1,2),N=1,NC)
1503  FORMAT(/10X,'CHECK FKV: '/
      130((5X,2E20.8)/))
      WRITE(NOFF,1506)(FNO(N),N=1,NC)
1506  FORMAT('   OF PIXELS IN EACH CLUSTER:'/
      1       30(/F12.1))
C     ********
```

```
      CALL DISTAN(NC,DIS)
C     ********
      DIST(KK)=DIS
      WRITE(7,1504)DIST(KK),KK
      WRITE(NOFF,1504)DIST(KK),KK
1504  FORMAT(/' IN KERVEC: DIST(KK)=',E20.8,' KK=',I2/)
      IF(KK.EQ.1)GO TO 500
      RATIO=(DIST(KK)-DIST(KK-1))/DIST(KK-1)
      WRITE(NOFF,1005)RATIO
      IF(ABS(RATIO).LT.0.001) GOTO 900
500   CONTINUE
900   CONTINUE
1005  FORMAT(' RATIO IN KERVEC:',E20.8)
1505  FORMAT(' COLOR:',64I1)
      RETURN
      END
C     ----------------------------------------------
C
C     USE CURRENT KERNEL VECTORS TO CALCULATE THE TOTAL
C     DISTANCE OF THE IMAGE
C     NC: NUMBER OF CLUSTERS
C     DISTOT: ( RESULT ) TOTAL DISTANCE
C
      SUBROUTINE DISTAN(NC,DISTOT)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
      COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
      COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
      DISTOT=0.
      REWIND NOF1
      REWIND NOF2
      REWIND NOFB
      DO 200 I=1,NOL
      INDXB=I
      READ(NOFB'INDXB)(ICOLOR(K),K=1,NOP)
      INDX1=I
      READ(NOF1'INDX1)(IA(K),K=1,NOP)
      INDX2=I
      READ(NOF2'INDX2)(IB(K),K=1,NOP)
C     STORE FEATURE VECTOR IN WORKING VARIABLE X
      DO 10 K=1,NOP
      AB(K,1)=FLOAT(IA(K))
      AB(K,2)=FLOAT(IB(K))
10    CONTINUE
      DO 100 J=1,NOP
      NCLSR=ICOLOR(J)
      SUM=0.
      DO 30 L=1,2
      SUM=SUM+(AB(J,L)-FKV(NCLSR,L))**2
30    CONTINUE
      DISTOT=DISTOT+SQRT(SUM)
100   CONTINUE
200   CONTINUE
```

```
      WRITE(NOFF,1501)DISTOT
1501  FORMAT(/' IN DISTAN:  DISTOT    = ',E20.8/)
      RETURN
      END
C     ----------------------------------------------
C
C
C     USE THE FINAL KERNEL VECTORS TO CLASSIFY THE IMAGE
C     INTO CLUSTERS ( SUBREGIONS )
C
      SUBROUTINE CLASS(NC)
      REAL FIRST(60)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
      COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
      COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
      WRITE(NOFF,1505)
1505  FORMAT(/' IN CLASS:'/)
      WRITE(7,1501)((FKV(N,L),L=1,2),N=1,NC)
      WRITE(NOFF,1501)((FKV(N,L),L=1,2),N=1,NC)
1501  FORMAT(10X,' FINAL KERNEL VECTERS :'/(30(5X,2E20.8/)))
1509  FORMAT(1X,64I1)
C     USE FINAL KERNAL VECTORS TO CLASSIFY THE PICTURE
C     NOFC: INTEGER OUTPUT; NOFD: REAL OUTPUT
      REWIND NOF1
      REWIND NOF2
      REWIND NOFB
      REWIND NOFC
C     RANGE OF FINAL RESULT 0..180
      FACT=180./FLOAT(NC)
      DO 200 I=1,NOL
      INDX1=I
      INDX2=I
      READ(NOF1'INDX1)(IA(K),K=1,NOP)
      READ(NOF2'INDX2)(IB(K),K=1,NOP)
      DO 10 J=1,NOP
      AB(J,1)=FLOAT(IA(J))
      AB(J,2)=FLOAT(IB(J))
10    CONTINUE
      DO 100 J=1,NOP
      DO 40 N=1,NC
      SUM=0.
      DO 30 L=1,2
      SUM=SUM+(AB(J,L)-FKV(N,L))**2
30    CONTINUE
      FIRST(N)=SQRT(SUM)
40    CONTINUE
C     DISTANCES TO FINAL KERNEL VECTOS OF NC CLUSTERS
C     FROM CURRENT PIXEL ARE STORED IN DIS ARRAY,
C     CALLING SUBROUTINE DISMIN TO FIND TO WHICH CLUSTER
C     THE CURRENT PIXEL IS CLOSER ( MINIMUM DISTANCE ).
C     THE RESULT IS KMIN-TH CLUSTER
      CALL DISMIN(FIRST,SMIN,NMIN,NC)
```

```
C       BLACK AND WHITE DISPLAY PURPOSE: NEEDS TO MULTIPLY A FACTOR
C       TO BE IN REASONABLE GRAY LEVEL RANGE
        IBW(J)=INT(FLOAT(NMIN)*FACT)
C       COLOR DISPLAY PURPOSE: AN INTEGER IN RANGE 1 TO 7
        ICOLOR(J)=NMIN
100     CONTINUE
        INDXB=I
        WRITE(NOFB'INDXB)(IBW(K),K=1,NOP)
        INDXC=I
        WRITE(NOFC'INDXC)(ICOLOR(K),K=1,NOP)
        IF (I.LE.64) WRITE(7,1509)(ICOLOR(K),K=1,64)
        IF (I.LE.64) WRITE(NOFF,1509)(ICOLOR(K),K=1,64)
200     CONTINUE
        RETURN
        END
C
C       ------------------------------------------------
C
        SUBROUTINE DISMIN(DARRY,DATMIN,NOMIN,NCLSTR)
C       PASS DARRY ARRAY WITH NCLSTR ELEMENTS MEANINGFUL
C       SEARCH FOR THE MINIMUM ELEMENT, NOMIN-TH ELEMENT,
C       WITH VALUE DATMIN; PASS BACK DATMIN AND NOMIN BACK
C       CALLING ROUTINE
        REAL DARRY(60)
        DATMIN=DARRY(1)
        NOMIN=1
C       ASSUME THE FIRST ELEMENT IS THE MINIMUM
C       THEN GO THROUGH THE REST OF THE ARRAY TO FIND ANY SMALLER
        IF (NCLSTR.EQ.1) GOTO 900
        DO 100 I=2,NCLSTR
        IF (DARRY(I).GE.DATMIN) GOTO 100
        DATMIN=DARRY(I)
        NOMIN=I
100     CONTINUE
900     CONTINUE
        RETURN
        END
```

APPENDIX SPACLR.FOR

```
C     ------------------------------------------------------------
----
C     9A123456789B123456789C123456789D123456789E123456789F12345678
9G12
C     FILE NAME: SPACLR.FOR
C     AUTO SPATIAL CLUSTERING FOR COLOR IMAGE DATA
C
C     REFERENCE FUKADA," SPATIAL CLUSTERING PROCEDURESFOR REGION
C             ANALYSIS",PATTERN RECOGNITION,12,395-403,(1980).
C
C
C     INPUTS: NOF1, FTN1.DAT, FEATURE 1
C             NOF2, FTN2.DAT, FEATURE 2
C             NOF3, FTN3.DAT, FEATURE 3
C     OUTPUTS:NOFC, FTN12.DAT, CLUSTERING RESULT FOR COLOR
DISPLAY
C             NOFD, FTN11.DAT, CLUSTERING RESULT FOR BLACK/WHITE
C                              DISPLAY
C             NOFF, FTN15.DAT, SOME PARAMETERS DURONG PROCESSING
C     SIZE OF IMAGE IS RESTICTED TO 256 BY 256 IN FEATURE SPACE
C     ------------------------------------------------------------
----
      LOGICAL*1 DDMMY(9)
      REAL PREV(60,3),PRENO(60)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
      COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
      COMMON /BLOCK4/RGBM(2,128,3),TRACE(128)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
      COMMON /BLOCK6/R(256,2),G(256,2),B(256,2)
      DATA NOF1,NOF2,NOF3,NOFC,NOFD,NOFF/1,2,3,12,11,15/
1701  FORMAT(' RUNNING PROGRAM SPACLR.FOR AUTO SPATIAL
CLUSTERING'/
     1        ' INPUTS , OUTPUTS: NOF1,NOF2,NOF3,NOFC,NOFD,NOFF'/
     6        ' FOR CHECK:',6I5/
     2        ' NOFC , NOFD STORE TEMPORARY DATA DURING
PROCESSING'/
     3        ' OUTPUTS: NOFC(REWRITTEN) 7 OR LESS COLORS'/
     4        '          NOFD(REWRITTEN) BLACK AND WHITE
DISPLAY'/
     5        '          NOFF, INFORMATIONS DURING PROCESSING')
1515  FORMAT('  MERGE ITERATION',I5)
1520  FORMAT(' THE ',I5,'-TH ITERATION REACHES MAXIMUM NO.
CLUSTERS'/
     1        ' NO. OF CLUSTERS:',I5)
1511  FORMAT(' KERNEL CANDIDATE VECTORS FOR',I4,' STARTING ',
     1        'CLUSTER CENTERS'/)
1512  FORMAT(' ENTER IMAGE DATA FILE SIZE ( FEATURE SPACE )'/
     1        ' NOL, NOP: FORMAT(2I5)')
1513  FORMAT(2I5)
1514  FORMAT(' CHECK:NOL,NOP,NOL2,NOP2,NOLH,NOPH'/6I5)
1516  FORMAT(' ENTER OPTIONS: (IOP(K),K=1,10)'/
     1        '  IOP(1): CONTROLS PRINTER, 1: MEANS , TRACE
MATRIX'/
```

```
     2        '  IOP(2): K-MEANS ALGORITHM DETAILS ON SCREEN'/
     3        '  IOP(3): MERGE DETAILS, LABEL(2,K) ARRAY'/
     4        '  IOP(4): NSTEP, NO. OF STEPS'/
     5        '  IOP(5): 1, SKIP K-MEAN ITERATION'/)
1517  FORMAT(10I6)
1518  FORMAT(' TODAY IS ',9A1)
      REWIND NOFF
      WRITE(7,1501)
      WRITE(NOFF,1501)
      CALL DATE(DDMMY)
      WRITE(NOFF,1518)(DDMMY(K),K=1,9)
      WRITE(7,1518)   (DDMMY(K),K=1,9)
      WRITE(NOFF,1701)NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
      WRITE(7,1701)NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
      WRITE(7,1512)
      WRITE(NOFF,1512)
      READ(5,1513)NOL,NOP
      WRITE(7,1516)
      WRITE(NOFF,1516)
      READ(5,1517)(IOP(K),K=1,10)
      WRITE(NOFF,1517)(IOP(K),K=1,10)
      WRITE(7,1517)(IOP(K),K=1,10)
      NOL2=NOL+NOL
      NOP2=NOP+NOP
      NOLH=NOL/2
      NOPH=NOP/2
      WRITE(NOFF,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
      WRITE(7,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
      DEFINE FILE NOF1(NOL,NOP,U,INDX1)
      DEFINE FILE NOF2(NOL,NOP,U,INDX2)
      DEFINE FILE NOF3(NOL,NOP,U,INDX3)
      DEFINE FILE NOFC(NOL,NOP,U,INDXC)
      DEFINE FILE NOFD(NOL,NOP,U,INDXD)
C     NOFF=FTN15.DAT UNFORMATTED
      REWIND NOF1
      REWIND NOF2
      REWIND NOF3
      REWIND NOFC
      REWIND NOFD
1501  FORMAT(' THIS IS THE LOG FILE OF EXECUTING SPACLR.FOR')
C     ---------------------------------------------
C     CALCULATE MEANS OF FEATURE VECTORS OF 2 BY 2 SUBIMAGE
C     STORE IN NOFC AND HALF NOFD 128 BY 128 SUBIMAGES
C     IN NOFC: FIRST HALF IS R MEANS, SECOND HALF IS G MEANS
C     IN NOFD: FIRST HALF IS B MEANS, SECOND HALF IS TRACES
      CALL DISPER
C     ---------------------------------------------
C     FIND MAX , MIN OF TRACE MATRIX
      CALL MAXMIN(DMAX,DMIN)
C     ---------------------------------------------
C     MERGING SECTION
      NSTEP=IOP(4)
      STEP=(DMAX-DMIN)/FLOAT(NSTEP)
      WRITE(7,1522)DMAX,DMIN,STEP
      WRITE(NOFF,1522)DMAX,DMIN,STEP
```

```
1522  FORMAT(' DMAX=',E20.8,' DMIN=',E20.8,' STEP=',E20.8)
      IPREV=0
C     ITERATIONS TO FIND MAXIMUM NO. OF CLUSTERS
      DO 300 I=1,NSTEP
      IM=I
      WRITE(NOFF,1515)IM
      CALL MERGE(IM,STEP,DMIN,NCLSR)
C     ACCEPTED NO. OF CLUSTERS: 7 OR LESS
      IF (IPREV.LE.7.AND.IPREV.GT.1) GOTO 333
      IPREV=NCLSR
C     SAVE CURRENT NUMBER OF CLUSTERS AND KERNEL VECTORS
      DO 200 J=1,IPREV
      PRENO(J)=CNO(J)
      PREV(J,1)=CKV(J,1)
      PREV(J,2)=CKV(J,2)
      PREV(J,3)=CKV(J,3)
200   CONTINUE
300   CONTINUE
333   NI=IM-1
      WRITE(NOFF,1520)NI,IPREV
      DO 350 J=1,IPREV
      CNO(J)=PRENO(J)
      CKV(J,1)=PREV(J,1)
      CKV(J,2)=PREV(J,2)
      CKV(J,3)=PREV(J,3)
350   CONTINUE
      WRITE(NOFF,1561)
1561  FORMAT(' MERGE ENDED WITH MAXIMUM NO. CLUSTERS')
      WRITE(NOFF,1562)((CKV(N,L),L=1,3),N=1,IPREV)
1562  FORMAT(' BEFORE SORTING'/
      1      ' KERNEL CANDIDATE VECTORS'/
      2      30((5X,3E20.8)/))
C     SORT THE CANDIDATE VECTORS
      NC=IPREV
C     ------------------------------------------------
C     SORT THE CANDIDATE KERNEL VECTORS
      CALL SORT(NC)
      WRITE(NOFF,1563)((CKV(N,L),L=1,3),N=1,NC)
1563  FORMAT(' SORTED KERNEL CANDIDATE VECTORS'/
      1      30((5X,3E20.8)/))
      IF (NC.GT.7) NC=7
C     FOR THE PURPOSE OF AED-512 PSEUDO COLOR DISPLAY
      WRITE(NOFF,1511)NC
C     IF (IOP(5).EQ.1) SKIP THE K-MEAN ITERATIONS
C     DIRECTLY USE MERGING RESULT CADIDATE KERNEL VECTORS
C     TO CLASSIFY THE IMAGE
      WRITE(7,1568)IOP(5)
1568  FORMAT(' IOP(5)=',I5)
      IF (IOP(5).NE.1) GOTO 700
      WRITE(7,1570)
1570  FORMAT(' SKIP K-MEAN ITERATION')
      DO 650 N=1,NC
      DO 640 L=1,3
      FKV(N,L)=CKV(N,L)
640   CONTINUE
```

```
650     CONTINUE
        GOTO 800
700     CONTINUE
        WRITE(7,1580)
1580    FORMAT(' CALLING KERVEC: K-MEAN ITERATION')
C       ----------------------------------------------------
C       ITERATIONS TO FIND MORE ACCURATE KERNEL VECTORS
C       CALLED FINAL KERNEL VECTORS
        CALL KERVEC(NC,KK,DD)
        WRITE(NOFF,1500)KK,DD
1500    FORMAT(1X,'CLUSTERING REPEATS',1X,I3,1X,'TIMES'/1X,
        1'THE FINAL WITHIN-CLASS DISTANCE IS',1X,E20.8/)
800     CONTINUE
C       -----------------------------------------------------
C       CLASSIFICATION SECTION
C       OUTPUTS: NOFC, COLOR DISPLAY RESULT
C                NOFD, BLACK/WHITE DISPLAY RESULT
        CALL CLASS(NC)
        WRITE(NOFF,1523)
1523    FORMAT(10X,'!!! COMPLETE EXECUTION OF PROGRAM SPACLR !!!')
        CALL EXIT
        END
C
C
C       ---------- SUBPROGRAMS ---------------
C
C       ----------------------------------------------------
C       SUBROUTINE TO CALCULATE TRACE MATRICES OF FEATURE MATRICES
C       STORED IN NOFD
        SUBROUTINE DISPER
        COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
        COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
        COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
        COMMON /BLOCK4/RGBM(2,128,3),TRACE(128)
        COMMON /BLOCK6/R(256,2),G(256,2),B(256,2)
        REWIND NOF1
        REWIND NOF2
        REWIND NOF3
        REWIND NOFC
        REWIND NOFD
C       PROCESS THROUGH ROWS OF DATA MATRIX
        DO 100 I=1,NOLH
        I2=I+I-1
        INDX1=I2
        INDX2=I2
        INDX3=I2
        DO 40 JJ=1,2
C       READ 2 LINES OF EACH FILE
        READ(NOF1'INDX1)(IR(K),K=1,NOP)
        DO 10 J=1,NOP
10      R(J,JJ)=FLOAT(IR(J))
        READ(NOF2'INDX2)(IG(K),K=1,NOP)
        DO 20 J=1,NOP
20      G(J,JJ)=FLOAT(IG(J))
        READ(NOF3'INDX3)(IB(K),K=1,NOP)
```

```
      DO 30 J=1,NOP
30    B(J,JJ)=FLOAT(IB(J))
40    CONTINUE
C     CALCULATION THROUGH EACH SUBIMAGE
      DO 80 K=1,NOPH
      K1=K+K-1
      K2=K1+1
      S1=0.
      S2=0.
      S3=0.
      DO 62 M=K1,K2
      DO 60 L=1,2
      S1=S1+R(M,L)
      S2=S2+G(M,L)
      S3=S3+B(M,L)
60    CONTINUE
62    CONTINUE
      RGBM(2,K,1)=S1*0.25
      RGBM(2,K,2)=S2*0.25
      RGBM(2,K,3)=S3*0.25
      S1=0.
      S2=0.
      S3=0.
      DO 72 M=K1,K2
      DO 70 L=1,2
      S1=S1+(R(M,L)-RGBM(2,K,1))**2
      S2=S2+(G(M,L)-RGBM(2,K,2))**2
      S3=S3+(B(M,L)-RGBM(2,K,3))**2
70    CONTINUE
72    CONTINUE
      TRACE(K)=(S1+S2+S3)*0.25
80    CONTINUE
      INDXC=I
      IF (IOP(1).EQ.1) WRITE(6,1001)(RGBM(2,K,1),K=1,32)
      WRITE(NOFC'INDXC)(RGBM(2,K,1),K=1,NOPH)
      INDXC=I+NOLH
      IF (IOP(1).EQ.1) WRITE(6,1002)(RGBM(2,K,2),K=1,32)
      WRITE(NOFC'INDXC)(RGBM(2,K,2),K=1,NOPH)
      INDXD=I
      IF (IOP(1).EQ.1) WRITE(6,1003)(RGBM(2,K,3),K=1,32)
      WRITE(NOFD'INDXD)(RGBM(2,K,3),K=1,NOPH)
      INDXD=I+NOLH
      IF (IOP(1).EQ.1) WRITE(6,1004)(TRACE(K),K=1,32)
      WRITE(NOFD'INDXD)(TRACE(K),K=1,NOPH)
100   CONTINUE
1001  FORMAT(' RM',32F4.0)
1002  FORMAT(' GM',32F4.0)
1003  FORMAT(' BM',32F4.0)
1004  FORMAT(' TR',32F4.0)
      RETURN
      END
C     -----------------------------------------------------
C     READ TRACE MATRIX TO FIND MAX , MIN
C
      SUBROUTINE MAXMIN(DMAX,DMIN)
```

```
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK4/RGBM(2,128,3),TRACE(128)
      REWIND NOFD
      INDXD=1+NOLH
      READ(NOFD'INDXD)(TRACE(K),K=1,NOPH)
      DMAX=TRACE(1)
      DMIN=TRACE(1)
      DO 10 J=2,NOPH
      IF (TRACE(J).LT.DMIN) DMIN=TRACE(J)
      IF (TRACE(J).GT.DMAX) DMAX=TRACE(J)
10    CONTINUE
      DO 100 I=2,NOLH
      INDXD=I+NOLH
      READ(NOFD'INDXD)(TRACE(K),K=1,NOPH)
      DO 30 J=1,NOPH
      IF (TRACE(J).LT.DMIN) DMIN=TRACE(J)
      IF (TRACE(J).GT.DMAX) DMAX=TRACE(J)
30    CONTINUE
100   CONTINUE
      WRITE(7,1001)DMAX,DMIN
1001  FORMAT(/' DMAX=',F12.4,' DMIN=',F12.4)
      RETURN
      END
C
C     ------------------------------------------
C     MERGE AND DECIDE KERNEL CANDIDATE VECTORS
C
      SUBROUTINE MERGE(IMRGE,DSTEP,DMIN,LLBS)
      REAL FIRST(60)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
      COMMON /BLOCK4/RGBM(2,128,3),TRACE(128)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1502  FORMAT(30(/' CNO(N):',F12.1))
1503  FORMAT(' *FMIN, DMRGE:',2E20.8)
1504  FORMAT(/' IMRGE:',I5)
1505  FORMAT(' FIRST SUBIMAGE:')
1506  FORMAT(' JOIN CLUSTER NL:')
1507  FORMAT(' NEW CLUSTER:')
1508  FORMAT(' LLBS.GE.60')
1532  FORMAT(15X,'THETA',7X,' SIGMA-SQUARE',3X,'
MERGING-DISTANCE:'/
      1        3E20.8)
1533  FORMAT(' K,LLBS,TRACE(K),THETA,DMRGE:'/2I5,3E20.8)
      REWIND NOFC
      REWIND NOFD
C     FOR EACH ITERATION: ZERO OUT THE VARIABLES
      DO 20 J=1,60
      CNO(J)=0.
      DO 10 L=1,3
      CKV(J,L)=0.
10    CONTINUE
20    CONTINUE
```

APPENDIX  SPACLR.FOR

```
      LLBS=0
C     DMIN: SIGMA-SQUARE
C     THETA:  SOME THETA
C     DMRGE:    MERGING DISTANCE
      THETA=DMIN+DSTEP*FLOAT(IMRGE)
      DMRGE=SQRT(4./3.*(THETA-DMIN))
      WRITE(NOFF,1532)THETA,DMIN,DMRGE
      WRITE(7,1504)IMRGE
C     GO THROUGH SUBIMAGES AND LABEL THEM WITH CLUSTERS
      DO 300 J=1,NOLH
      IF (J.GT.1) GOTO 35
C     J=1 : CASE OF FIRST LINE OF SUBIMAGES
      DO 30 K=1,NOPH
      LABEL(2,K)=0
      DO 30 L=1,3
30    RGBM(1,K,L)=0.
      GOTO 45
35    CONTINUE
C     GET PREVIOUS LINE OF SUBIMAGES IN RGBM ARRAY
      DO 40 K=1,NOPH
      DO 40 L=1,3
40    RGBM(1,K,L)=RGBM(2,K,L)
45    CONTINUE
      INDXC=J
      READ(NOFC'INDXC)(RGBM(2,K,1),K=1,NOPH)
      INDXC=J+NOLH
      READ(NOFC'INDXC)(RGBM(2,K,2),K=1,NOPH)
      INDXD=J
      READ(NOFD'INDXD)(RGBM(2,K,3),K=1,NOPH)
C     INTIAL LABEL FOR EACH SUBIMAGE
      DO 50 K=1,NOPH
      LABEL(1,K)=LABEL(2,K)
      LABEL(2,K)=0
50    CONTINUE
      INDXD=J+NOLH
      READ(NOFD'INDXD)(TRACE(K),K=1,NOPH)
C     GO THROUGH IMAGES ONE BY ONE
      DO 201 K=1,NOPH
      IF (IDP(1).EQ.9) WRITE(7,1533)K,LLBS,TRACE(K),THETA,DMRGE
      IF (LLBS.GE.60) WRITE(7,1508)
      IF (LLBS.GE.60) GOTO 900
C     CHECK IF THE TRACE OF CURRENT SUBIMAGE > THETA
      IF (TRACE(K).GT.THETA) GOTO 200
C     SKIP
      IF (J.GT.1) GOTO 52
C     J=1: FIRST LINE OF SUBIMAGES
C     THE FIRST LINE SECTION: CONSIDERING THE NEIGHBOR
      M1=2
      M2=2
      K1=K-1
      K2=K
      GOTO 54
52    CONTINUE
C     NOT THE FIRST LINE; SO PREVIOUS LINE EXISTS
      M1=1
```

```
      M2=2
      K1=K
      K2=K
54    CONTINUE
C     CHECK IF FIRST SUBIMAGE OR NOT
      IF (LLBS.EQ.0) GOTO 90
      IF (LABEL(M1,K1).EQ.0) GOTO 55
C     POTENTIAL NEIGHBOR NOT LABELLED, DIRECTLY CHECK CLUSTERS
C     LABEL(M1,K1) NEIGHBOR HAS BEEN LABELLED
C     AND SPATIAL CLUSTERING SHOULD BE APPLIED
      DIFF=0.
      DO 62 L=1,3
62    DIFF=DIFF+(RGBM(M1,K1,L)-RGBM(M2,K2,L))**2
      DIFF=SQRT(DIFF)
      IF (DIFF.GT.DMRGE) GOTO 55
C     WITHIN MERGING DISTANCE ?
      LABEL(M2,K2)=LABEL(M1,K1)
      NL=LABEL(M1,K1)
      LABEL(M2,K2)=NL
      CNO(NL)=CNO(NL)+1.
      DO 64 L=1,3
64    CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+RGBM(M2,K2,L))/CNO(NL)
      GOTO 200
55    CONTINUE
      DO 58 N=1,LLBS
      DIFF=0.
      DO 56 L=1,3
      DIFF=DIFF+(CKV(N,L)-RGBM(2,K2,L))**2
56    CONTINUE
      FIRST(N)=SQRT(DIFF)
58    CONTINUE
      CALL DISMIN(FIRST,FMIN,NL,LLBS)
      IF (FMIN.GT.DMRGE) GOTO 90
      IF (IOP(1).EQ.9) WRITE(7,1503)FMIN,DMRGE
C     LABEL CURRENT SUBIMAGE WITH CLOSEST CENTER
      LABEL(M2,K2)=NL
C     UPDATE NO. OF SUBIMAGES OF CURRENT CLUSTER
      CNO(NL)=CNO(NL)+1.
C     UPDATE MEAN VECTOR OF THIS CLUSTER
      DO 60 L=1,3
      CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+RGBM(2,K2,L))/CNO(NL)
60    CONTINUE
      IF (IOP(1).EQ.9) WRITE(7,1506)
      GOTO 200
C     NEW CLUSTER SECTION
90    CONTINUE
      LLBS=LLBS+1
C     UPDATE   OF SUBIMAGES OF THIS CLUSTER
      CNO(LLBS)=CNO(LLBS)+1.
C     UPDATE NEW CLUSTER VECTOR
      DO 92 L=1,3
92    CKV(LLBS,L)=RGBM(M2,K,L)
200   CONTINUE
201   CONTINUE
250   CONTINUE
```

```
C       CHECK CURRENT LINE'S LABELS
        IF (IOP(3).EQ.1) WRITE(7,1545)(LABEL(2,K),K=1,32)
300     CONTINUE
900     CONTINUE
        IF (IOP(1).EQ.9) WRITE(7,1502)(CNO(N),N=1,LLBS)
        IF (IOP(1).EQ.9) WRITE(7,1501)IMRGE,LLBS
        WRITE(NOFF,1501)IMRGE,LLBS
1545    FORMAT(' LABEL',32I2)
1501    FORMAT(/' MERGE ITERATION:',I5,' END WITH LLBS: ',I5/)
        RETURN
        END
C
C       ------------------------------------------------------------
C       SORTING THE KERNEL VECTORS
C
        SUBROUTINE SORT(NCLRS)
        REAL TEMP(3)
        COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
        COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
        DO 30 I=2,NCLRS
        I2=NCLRS+1-I
        DO 20 J=1,I2
        IF (CKV(J+1,1).GE.CKV(J,1)) GOTO 20
        DO 10 L=1,3
        TEMP(L)=CKV(J+1,L)
        CKV(J+1,L)=CKV(J,L)
        CKV(J,L)=TEMP(L)
10      CONTINUE
20      CONTINUE
30      CONTINUE
        WRITE(NOFF,1533)
1533    FORMAT(/' SORTING CANDIDATE KERNEL VECTORS'/)
        RETURN
        END
C
C       --------------------------------------------------------
C       TO FIND FINAL KERNEL VECTORS
C       LIMIT TO 10 ITERATIONS
C
        SUBROUTINE KERVEC(NC,KK,DIS)
C       DIST ARRAY STORES THE TOTAL DISTANCES OF ITERATIONS
C       C ARRAY STORES NUMBER OF PIXELS FOR EACH CLUSTER
C       D ARRAY STORES TEMPORARY DISTANCES TO CLUSTER CENTERS
C          FOR CURRENT PIXEL BEING PROCESSED
        REAL DIST(10),FIRST(60)
        COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
        COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
        COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
        COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
        COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1020    FORMAT(' IN KERVEC, KK:',I5)
C       FINAL KERNELVECTORS SAVED IN FKV ARRAY
C       FNO STORE NO. OF PIXELS IN EACH CLUSTER
C       K-MEANS ALGORITHM
C       10-JUN-82 CORRECT IMPLEMENTATION
```

APPENDIX SPACLR.FOR

```
C       REFERENCE: TOU AND GONZALEZ," PATTERN RECOGNITION
C        PRINCIPLES", PP.94-97
        DO 12 N=1,NC
        FNO(N)=CNO(N)
        CNO(N)=0.
        DO 10 L=1,3
        FKV(N,L)=CKV(N,L)
        CKV(N,L)=0.
10      CONTINUE
12      CONTINUE
C       NO. OF ITERATIONS LIMIT TO 10
        DO 500 KK=1,10
        WRITE(7,    1020)KK
        WRITE(NOFF,1020)KK
        DO 15 N=1,NC
15      FNO(N)=0.
        REWIND NOF1
        REWIND NOF2
        REWIND NOF3
        REWIND NOFD
C       FOR EACH ITERATION:
C       REWIND THE FEATURES FILES: R , G , B COMPONENTS
C       REWIND THE TEMPORARY CLASSIFIED RESULT FILE, NOFD
C       CLASSIFYING STANDARDS IN FKV ARRAY
C       AT THE SAME TIME, COLLECTING THE NEW CENTERS IN CKV ARRAY
C       I.E., UPDATING THE KERNEL VECTORS BY CURRENT CLUSTERING
C       ICOLOR ARRAY STORES CLASSIFIED RESULT OF CURRENT LINE
        DO 200 I=1,NOL
        INDX1=I
        READ(NOF1'INDX1)(IR(K),K=1,NOP)
        INDX2=I
        READ(NOF2'INDX2)(IG(K),K=1,NOP)
        INDX3=I
        READ(NOF3'INDX3)(IB(K),K=1,NOP)
        DO 20 J=1,NOP
        RGB(J,1)=FLOAT(IR(J))
        RGB(J,2)=FLOAT(IG(J))
        RGB(J,3)=FLOAT(IB(J))
20      CONTINUE
C       GO THROUGH PIXELS TO LABEL THEM WITH CLUSTERS
        DO 100 J=1,NOP
        DO 40 N=1,NC
        SUM=0.
        DO 30 L=1,3
30      SUM=SUM+(RGB(J,L)-FKV(N,L))**2
40      FIRST(N)=SQRT(SUM)
        CALL DISMIN(FIRST,FMIN,NN,NC)
        ICOLOR(J)=NN
        CNO(NN)=CNO(NN)+1.
C       CURRENT PIXEL WAS FOUND CLOSER TO NN-TH CLUSTER
C       THE FEATURES SHOULD BE INCLUDED TO UPDATE THE NN-TH
C       KERNEL VECTOR
        DO 70 L=1,3
70      CKV(NN,L)=(CKV(NN,L)*(CNO(NN)-1.)+RGB(J,L))/CNO(NN)
100     CONTINUE
```

```
      INDXD=I
      IF (IOP(2).EQ.1) WRITE(7,1505)(ICOLOR(K),K=1,64)
      WRITE(NOFD'INDXD)(ICOLOR(K),K=1,NOP)
200   CONTINUE
C     CURRENT IN CKV; STORE IN FKV TO BE USED TO CLASSIFY
C     IN ROUTINE DISTAN, AND WILL GIVE TOTAL DISTANCE
      DO 260 N=1,NC
      FNO(N)=CNO(N)
      CNO(N)=0.
      DO 250 L=1,3
      FKV(N,L)=CKV(N,L)
      CKV(N,L)=0.
250   CONTINUE
260   CONTINUE
      WRITE(NOFF,1503)((FKV(N,L),L=1,3),N=1,NC)
1503  FORMAT(/10X,'CHECK FKV: '/
      130((5X,3E20.8)/))
      WRITE(NOFF,1506)(FNO(N),N=1,NC)
1506  FORMAT('   OF PIXELS IN EACH CLUSTER:'/
      1       30(/F12.1))
C     ********
      CALL DISTAN(NC,DIS)
C     ********
      DIST(KK)=DIS
      WRITE(7,1504)DIST(KK),KK
      WRITE(NOFF,1504)DIST(KK),KK
1504  FORMAT(/' IN KERVEC: DIST(KK)=',E20.8,' KK=',I2/)
      IF(KK.EQ.1)GO TO 500
      RATIO=(DIST(KK)-DIST(KK-1))/DIST(KK-1)
      WRITE(NOFF,1005)RATIO
      IF(ABS(RATIO).LT.0.001) GOTO 900
500   CONTINUE
900   CONTINUE
1005  FORMAT(' RATIO IN KERVEC:',E20.8)
1505  FORMAT(' COLOR:',64I1)
      RETURN
      END
C     ----------------------------------------------
C
C     USE CURRENT KERNEL VECTORS TO CALCULATE THE TOTAL
C     DISTANCE OF THE IMAGE
C     NC: NUMBER OF CLUSTERS
C     DISTOT: ( RESULT ) TOTAL DISTANCE
C
      SUBROUTINE DISTAN(NC,DISTOT)
      COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
      COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
      DISTOT=0.
      REWIND NOF1
      REWIND NOF2
      REWIND NOF3
      REWIND NOFD
```

```
      DO 200 I=1,NOL
      INDXD=I
      READ(NOFD'INDXD)(ICOLOR(K),K=1,NOP)
      INDX1=I
      READ(NOF1'INDX1)(IR(K),K=1,NOP)
      INDX2=I
      READ(NOF2'INDX2)(IG(K),K=1,NOP)
      INDX3=I
      READ(NOF3'INDX3)(IB(K),K=1,NOP)
C     STORE FEATURE VECTOR IN WORKING VARIABLE X
      DO 10 K=1,NOP
      RGB(K,1)=FLOAT(IR(K))
      RGB(K,2)=FLOAT(IG(K))
      RGB(K,3)=FLOAT(IB(K))
10    CONTINUE
      DO 100 J=1,NOP
      NCLSR=ICOLOR(J)
      SUM=0.
      DO 30 L=1,3
      SUM=SUM+(RGB(J,L)-FKV(NCLSR,L))**2
30    CONTINUE
      DISTOT=DISTOT+SQRT(SUM)
100   CONTINUE
200   CONTINUE
      WRITE(NOFF,1501)DISTOT
1501  FORMAT(/' IN DISTAN:  DISTOT     = ',E20.8/)
      RETURN
      END
C     ----------------------------------------------------
C
C
C     USE THE FINAL KERNEL VECTORS TO CLASSIFY THE IMAGE
C     INTO CLUSTERS ( SUBREGIONS )
C
      SUBROUTINE CLASS(NC)
      REAL FIRST(60)
      COMMON /BLOCK0/ IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
      COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
      COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
      COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
      COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
      WRITE(NOFF,1505)
1505  FORMAT(/' IN CLASS:'/)
      WRITE(7,1501)((FKV(N,L),L=1,3),N=1,NC)
      WRITE(NOFF,1501)((FKV(N,L),L=1,3),N=1,NC)
1501  FORMAT(10X,' FINAL KERNEL VECTERS :'/(30(5X,3E20.8/)))
1509  FORMAT(1X,64I1)
C     USE FINAL KERNAL VECTORS TO CLASSIFY THE PICTURE
C     NOFC: INTEGER OUTPUT; NOFD: REAL OUTPUT
      REWIND NOF1
      REWIND NOF2
      REWIND NOF3
      REWIND NOFC
      REWIND NOFD
C     RANGE OF FINAL RESULT 0..180
```

```
      FACT=180./FLOAT(NC)
      DO 200 I=1,NOL
      INDX1=I
      INDX2=I
      INDX3=I
      READ(NOF1'INDX1)(IR(K),K=1,NOP)
      READ(NOF2'INDX2)(IG(K),K=1,NOP)
      READ(NOF3'INDX3)(IB(K),K=1,NOP)
      DO 10 J=1,NOP
      RGB(J,1)=FLOAT(IR(J))
      RGB(J,2)=FLOAT(IG(J))
      RGB(J,3)=FLOAT(IB(J))
10    CONTINUE
      DO 100 J=1,NOP
      DO 40 N=1,NC
      SUM=0.
      DO 30 L=1,3
      SUM=SUM+(RGB(J,L)-FKV(N,L))**2
30    CONTINUE
      FIRST(N)=SQRT(SUM)
40    CONTINUE
C     DISTANCES TO FINAL KERNEL VECTOS OF NC CLUSTERS
C     FROM CURRENT PIXEL ARE STORED IN DIS ARRAY,
C     CALLING SUBROUTINE DISMIN TO FIND TO WHICH CLUSTER
C     THE CURRENT PIXEL IS CLOSER ( MINIMUM DISTANCE ).
C     THE RESULT IS KMIN-TH CLUSTER
      CALL DISMIN(FIRST,SMIN,NMIN,NC)
C     BLACK AND WHITE DISPLAY PURPOSE: NEEDS TO MULTIPLY A FACTOR
C     TO BE IN REASONABLE GRAY LEVEL RANGE
      IBW(J)=INT(FLOAT(NMIN)*FACT)
C     COLOR DISPLAY PURPOSE: AN INTEGER IN RANGE 1 TO 7
      ICOLOR(J)=NMIN
100   CONTINUE
      INDXD=I
      WRITE(NOFD'INDXD)(IBW(K),K=1,NOP)
      INDXC=I
      WRITE(NOFC'INDXC)(ICOLOR(K),K=1,NOP)
      IF (I.LE.64) WRITE(7,1509)(ICOLOR(K),K=1,64)
      IF (I.LE.64) WRITE(NOFF,1509)(ICOLOR(K),K=1,64)
200   CONTINUE
      RETURN
      END
C
C     ----------------------------------------------
C
      SUBROUTINE DISMIN(DARRY,DATMIN,NOMIN,NCLSTR)
C     PASS DARRY ARRAY WITH NCLSTR ELEMENTS MEANINGFUL
C     SEARCH FOR THE MINIMUM ELEMENT, NOMIN-TH ELEMENT,
C     WITH VALUE DATMIN; PASS BACK DATMIN AND NOMIN BACK
C     CALLING ROUTINE
      REAL DARRY(60)
      DATMIN=DARRY(1)
      NOMIN=1
C     ASSUME THE FIRST ELEMENT IS THE MINIMUM
C     THEN GO THROUGH THE REST OF THE ARRAY TO FIND ANY SMALLER
```

```
      IF (NCLSTR.EQ.1) GOTO 900
      DO 100 I=2,NCLSTR
      IF (DARRY(I).GE.DATMIN) GOTO 100
      DATMIN=DARRY(I)
      NOMIN=I
100   CONTINUE
900   CONTINUE
      RETURN
      END
```